Practitioner's Docket No. JP920000066US;954-010377-US (PAR) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Hurst et al. Group No.:
Serial No.: 0 /
Filed: Herewith Examiner:
For: DOCUMENT PROCESSING METHOD, SYSTEM AND MEDIUM

Assistant Commissioner for Patents
Washington, D.C. 20231

TRANSMITTAL OF CERTIFIED COPY

Attached please find the certified copy of the foreign application from which priority is claimed for this case:

Country: Japan

Application Number: 2000-190335

Filing Date: 23.06.2000

WARNING: *"When a document that is required by statute to be certified must be filed, a copy, including a photocopy or facsimile transmission of the certification is not acceptable." 37 C.F.R. 1.4(f) (emphasis added).*

David Aker
SIGNATURE OF PRACTITIONER

Reg. No. 29,277

David Aker
(type or print name of practitioner)

Tel. No. (203) 259-1800

Perman & Green, LLP

P.O. Address 425 Post Road
Fairfield, CT 06430

Customer No.: 2512

NOTE: The claim to priority need be in no special form and may be made by the attorney or agent, if the foreign application is referred to in the oath or declaration, as required by § 1.63.

CERTIFICATE OF MAILING (37 C.F.R. 1.8a)

I hereby certify that this correspondence is, on the date shown below is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Signature

Date: _____

(type or print name of person certifying)

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年　６月２３日

出 願 番 号
Application Number: 特願２０００－１９０３３５

出 願 人
Applicant (s): インターナショナル・ビジネス・マシーンズ・コーポレーション

２０００年１２月２２日

特 許 庁 長 官
Commissioner,
Patent Office 及 川 耕 造

出証番号　出証特２０００－３１０７２８７

【書類名】　特許願

【整理番号】　JP9000066

【提出日】　平成12年 6月23日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 17/21

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　マシュー・フランシス・ハースト

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　那須川　哲哉

【特許出願人】

【識別番号】　390009531

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【復代理人】

【識別番号】　100112520

【弁理士】

【氏名又は名称】　林　茂則

【電話番号】　046-277-0540

【選任した代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【選任した復代理人】

【識別番号】　100110607

【弁理士】

【氏名又は名称】　間山　進也

【選任した復代理人】

【識別番号】　100098121

【弁理士】

【氏名又は名称】　間山　世津子

【手数料の表示】

【予納台帳番号】　091156

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　文書処理方法、文書処理システムおよび媒体

【特許請求の範囲】

【請求項１】　文書処理システムにおいて空白文字またはタブその他のスペースで文字がレイアウトされている文書から意味のあるテキストブロックを抽出する文書処理方法であって、

前記文書から文字、記号、空白文字その他シンボルで構成されるオブジェクトを生成するステップと、

前記オブジェクト間の接続候補を生成するステップと、

前記接続候補の接続の妥当性を言語モデルを用いて評価するステップと、

を含む文書処理方法。

【請求項２】　前記接続候補の接続が妥当であると判断された場合には、前記接続候補の接続元および接続先の前記オブジェクトを結合するステップをさらに含む請求項１記載の文書処理方法。

【請求項３】　前記オブジェクトは、前記文書の空間位置を示す座標に関連付けられて生成される請求項１または２記載の文書処理方法。

【請求項４】　前記オブジェクトの結合により前記テキストブロックが生成され、前記テキストブロックは、前記オブジェクトを含む最小面積の方形領域で定義され、前記文書における前記方形領域の対角２点の座標で位置が特定される請求項３記載の文書処理方法。

【請求項５】　前記オブジェクト間の接続候補は、接続元オブジェクトの右方に隣接するオブジェクトとの接続、または、前記接続元オブジェクトが存在する行の次行に存在し、前記接続元オブジェクトより左方に位置する何れかのオブジェクトとの接続である請求項１～４の何れか一項に記載の文書処理方法。

【請求項６】　前記言語モデルはｎグラムモデルである請求項１～５の何れか一項に記載の文書処理方法。

【請求項７】　前記オブジェクトの生成ステップには、

前記文書の空間座標に関連付けて、前記座標毎のシンボルを取得するステップと、

前記文書の１行内の前記シンボルのタイプを判断し、１つまたは連続する文字、記号その他のキャラクタで構成されるトークン、または、１つまたは連続する空白文字で構成されるスペースを生成するステップと、

前記スペースの上下方向の隣接関係を判断し、複数行にわたるスペースで構成されるストリームを生成するステップと、

前記トークンと前記ストリームとの位置関係を判断し、前記トークンまたは空白文字を含む初期テキストブロックを生成するステップと、

を含む請求項１～６の何れか一項に記載の文書処理方法。

【請求項８】　前記トークンまたはスペースを生成するステップにおいて、前記タイプが空白文字でないと判断され、前記行内において隣接するシンボルの前記タイプが同じであると判断された場合には、前記シンボルを連続するキャラクタで構成される１つのトークンとして記録し、前記タイプが空白文字であると判断された場合には前記シンボルを１つまたは連続するスペースとして記録し、

前記ストリームを生成するステップにおいて、前記スペースが異なる行において上または下の方向に隣接すると判断された時には前記スペースをストリームとして記録し、

前記オブジェクトの生成ステップにおいて、１行内の２つのトークンに挟まれたスペースがストリームでない場合には、前記２つのトークンとその間のスペースを初期テキストブロックとして結合する請求項７記載の文書処理方法。

【請求項９】　前記初期テキストブロックとその接続候補の全てとを生成するステップと、

前記初期テキストブロックおよび接続候補の全てから単一要素の初期テキストブロックおよび接続候補を抽出するステップと、

前記単一要素の初期テキストブロックおよび接続候補の接続妥当性を言語モデルを用いて判断するステップと、

前記接続妥当性が妥当であると判断された時には、前記単一要素の初期テキストブロックを結合するステップと、

を含む請求項７または８記載の文書処理方法。

【請求項１０】　前記オブジェクト、初期テキストブロックまたはそれらが

結合されたテキストブロックとの間に単一の接続候補のみが存在する時には、言語モデルを用いた接続妥当性を判断することなくこれらを結合する請求項１～９の何れか一項に記載の文書処理方法。

【請求項１１】　空白文字またはタブその他のスペースで文字がレイアウトされている文書から意味のあるテキストブロックを抽出する文書処理システムであって、

前記文書から文字、記号、空白文字その他シンボルで構成されるオブジェクトを生成する手段と、

前記オブジェクト間の接続候補を生成する手段と、

前記接続候補の接続の妥当性を言語モデルを用いて評価する手段と、

を含む文書処理システム。

【請求項１２】　前記接続候補の接続が妥当であると判断された場合には、前記接続候補の接続元および接続先の前記オブジェクトを結合する手段をさらに含む請求項１１記載の文書処理システム。

【請求項１３】　前記オブジェクトは、前記文書の空間位置を示す座標に関連付けられて生成される請求項１１または１２記載の文書処理システム。

【請求項１４】　前記オブジェクトの結合により前記テキストブロックが生成され、前記テキストブロックは、前記オブジェクトを含む最小面積の方形領域で定義され、前記文書における前記方形領域の対角２点の座標で位置が特定される請求項１３記載の文書処理システム。

【請求項１５】　前記オブジェクト間の接続候補は、接続元オブジェクトの右方に隣接するオブジェクトとの接続、または、前記接続元オブジェクトが存在する行の次行に存在し、前記接続元オブジェクトより左方に位置する何れかのオブジェクトとの接続である請求項１１～１４の何れか一項に記載の文書処理システム。

【請求項１６】　前記言語モデルはｎグラムモデルである請求項１１～１５の何れか一項に記載の文書処理システム。

【請求項１７】　前記オブジェクトの生成手段には、

前記文書の空間座標に関連付けて、前記座標毎のシンボルを取得する手段と、

前記文書の１行内の前記シンボルのタイプを判断し、１つまたは連続する文字、記号その他のキャラクタで構成されるトークン、または、１つまたは連続する空白文字で構成されるスペースを生成する手段と、

前記スペースの上下方向の隣接関係を判断し、複数行にわたるスペースで構成されるストリームを生成する手段と、

前記トークンと前記ストリームとの位置関係を判断し、前記トークンおよび空白文字を含む初期テキストブロックを生成する手段と、

を含む請求項１１～１６の何れか一項に記載の文書処理システム。

【請求項１８】　前記トークンまたはスペースを生成する手段において、前記タイプが空白文字でないと判断され、前記行内において隣接するシンボルの前記タイプが同じであると判断された場合には、前記シンボルを連続するキャラクタで構成される１つのトークンとして記録し、前記タイプが空白文字であると判断された場合には前記シンボルを１つまたは連続するスペースとして記録し、

前記ストリームを生成する手段において、前記スペースが異なる行において上または下の方向に隣接すると判断された時には前記スペースをストリームとして記録し、

前記オブジェクトの生成手段において、１行内の２つのトークンに挟まれたスペースがストリームでない場合には、前記２つのトークンとその間のスペースを初期テキストブロックとして結合する請求項１７記載の文書処理システム。

【請求項１９】　前記初期テキストブロックとその接続候補の全てとを生成する手段と、

前記初期テキストブロックおよび接続候補の全てから単一要素の初期テキストブロックおよび接続候補を抽出する手段と、

前記単一要素の初期テキストブロックおよび接続候補の接続妥当性を言語モデルを用いて判断する手段と、

前記接続妥当性が妥当であると判断された時には、前記単一要素の初期テキストブロックを結合する手段と、

を含む請求項１７および１８記載の文書処理システム。

【請求項２０】　前記オブジェクト、初期テキストブロックまたはそれらが

結合されたテキストブロックとの間に単一の接続候補のみが存在する時には、言語モデルを用いた接続妥当性を判断することなくこれらを結合する請求項１１～１９の何れか一項に記載の文書処理システム。

【請求項２１】　空白文字またはタブその他のスペースで文字がレイアウトされている文書から意味のあるテキストブロックを抽出するプログラムが記録されたコンピュータ可読な記録媒体であって、前記プログラムは、

前記文書から文字、記号、空白文字その他のシンボルで構成されるオブジェクトを生成し、

前記オブジェクト間の接続候補を生成し、

前記接続候補の接続の妥当性を言語モデルを用いて評価し、

前記接続候補の接続が妥当であると判断された場合には、前記接続候補の接続元および接続先の前記オブジェクトを結合する手順をコンピュータに実行させるものである記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、多段組、表、箇条書き、タイトル付け等任意にレイアウトされた文書から意味のあるテキストブロックを抽出する文書処理方法、システムおよび媒体に関する。本発明の技術はテキストマイニング処理、機械翻訳等自然言語文書処理の前処理に適用し得る。

【０００２】

【従来の技術】

近年コンピュータネットワーク上で流通する電子化された文書あるいはスキャナ等の読み取り装置で電子化された文書が膨大に蓄積されており、これら文書の活用が望まれている。蓄積されている文書の活用方法の１つにテキストマイニング処理（文書の概要を自動的に把握し、内容の経時的変化や傾向を把握等する文書検索処理の一種）がある。また、機械翻訳の元データとして活用される場合がある。

【０００３】

これら蓄積文書の活用を考慮すれば、文書のレイアウトを解析する必要がある。一般に流通している文書たとえばホームページにアップロードされる文書等では人間が視覚により把握しやすいようにレイアウトされている。また、スキャナ等により電子化された文書データでは、原稿は紙媒体の文書であり、通常の印刷様式に基づいてレイアウトされている。これらレイアウトされた文書には、文章の本体である段落のほかに、タイトル、ヘッダ、リスト、表等が含まれ、また段落も２段組等多段で表示される場合が多い。さらに表の中には、横書きの要素ばかりでなく、縦書き要素が含まれる場合もある。このため、元文書のレイアウトを考慮しなければ満足な文書解析を自動的に行うことは困難である。

【０００４】

レイアウト解析の方法には、空間的な特徴に着目する方法がある。たとえば空白に着目し、空白行が挿入されている場合にはその後段は段落であると推定できる。

【０００５】

【発明が解決しようとする課題】

ところが、これら空間的な特徴により意味のあるテキストブロックを抽出するには限界がある。たとえば段落要素(文章がページの纏まった領域でタイプされているようなテキスト文書）の場合と表中のテキストの場合を比較すれば、各々空白の用い方が相違する。つまり行頭に空白文字（またはタブによる空白）が表示されている時には段落の始めであることが認められるが、表中の空白は通常そのようには配置されない。また、箇条書き等リスト表示される時には行頭にインデントが付されたり、行間に空白行が挿入される。これら多様にレイアウトされたテキスト文書を一元的に空白の有無のみで解析するのは困難である。

【０００６】

また、仮にレイアウトからテキストのブロックが抽出されても、そのブロック内の文章（あるいは単語の連なり）の意味上の評価が行われているわけではない。このため、特に表や見出し、リスト等段要素のように纏まったテキスト文書として表示されていない要素の場合にはブロックが分断され、その意味が正確に読み取れない。

【０００７】

ところで、蓄積されている文書の高度な利用（たとえばテキストマイニング）の場合には、文書の内容を自動的に判別する必要があるが、内容的に重要なメッセージは段落要素よりも表、リスト（箇条書き）等に含められることが多い。従来、空間的な特徴に基づくレイアウト解析の場合には、その解析の困難性から表、リスト（箇条書き）等の要素の解析を断念していた（あるいは要素が断片化されるため、その後の利用が困難であった）。しかしながら、むしろ後の高度利用を考慮すれば、これら表、リスト（箇条書き）等の要素にこそ重要なメッセージが内包されている可能性が高く、後の意味解析にまで適用し得る形態で抽出することが望まれる。

【０００８】

本発明の目的は、表、箇条書き、多段組等任意にレイアウトされた文書から意味のあるテキストブロックを抽出する技術を提供することにある。

【０００９】

【課題を解決するための手段】

本願の発明の概略を説明すれば、以下の通りである。すなわち、本発明は、空白文字等のスペースにより任意にレイアウトされた文書から、たとえば単語に代表されるトークン、１つまたは連続した空白文字からなるスペース、またはこれらの結合等文書を構成するオブジェクトを生成する。オブジェクトは文書の空間位置に関連付けて生成する。そしてオブジェクト間の接続候補を生成する。オブジェクトと接続候補はグラフ理論の点（ノード）と辺（弧）に対応付けることができる。各リンクの妥当性を言語モデル（たとえばNグラムモデル）により判断し、接続候補（リンク）が妥当であると判断されればオブジェクトを結合する。

【００１０】

このように文書を処理することにより、様々にレイアウトされた文書において意味のあるテキストブロックを効率的に抽出することが可能になる。

【００１１】

オブジェクト間に生成される接続候補（リンク）は、オブジェクトの右側（横書き文書の場合）の他のオブジェクトあるいは次行（横書き文書の場合）のそれ

より左側に位置するオブジェクトとの間に生成できる。

【００１２】

オブジェクトの生成にあたっては、１行内の複数トークンあるいはスペースを結合して初期テキストブロックを生成できる。この時、列方向に連続するスペースをストリームとして定義し、このストリームの文書内空間位置の情報を利用できる。つまり、ストリームで分断されるトークンは結合されず、ストリームで分断されないトークンは結合される。

【００１３】

トークンの生成は、文書内空間座標に対応付けて１文字ずつキャラクタを取得し、このキャラクタのタイプ（アルファベット文字であるか、記号であるか、空白文字であるか等）を判断できる。タイプが同じキャラクタが連続する場合には１つのトークンとして記録できる。空白が連続する場合にはスペースとして記録できる。

【００１４】

オブジェクト間の接続妥当性は、初期テキストブロックの接続妥当性で評価できる。また、この評価は複数段階行える。まず、複数のオブジェクト間の全ての接続候補（接続可能性）において、単一要素のサブクラスタ（単一の入出次数を有するオブジェクトの集合）を生成できる。そして、この単一要素サブクラスタにおける接続妥当性を言語モデルを用いて評価できる。接続が妥当であれば、この単一要素サブクラスタを構成するオブジェクト（初期テキストブロック）を結合（マージ）できる。次に、マージした後のテキストブロック間の接続妥当性を同様の手法により評価できる。このようにして効率的に全ての接続候補を評価することができる。なお、接続候補が単一の場合には、言語モデルを用いた評価を行うことなくその接続候補の接続元および接続先のオブジェクト（初期テキストブロック、マージしたテキストブロック）を結合できる。

【００１５】

【発明の実施の形態】

以下、本発明の実施の形態を図面に基づいて詳細に説明する。ただし、本発明は多くの異なる態様で実施することが可能であり、本実施の形態の記載内容に限

定して解釈すべきではない。なお、実施の形態の全体を通して同じ要素には同じ番号を付するものとする。

【００１６】

以下の実施の形態では、主に方法またはシステムについて説明するが、当業者であれば明らかなとおり、本発明は方法、システムの他、コンピュータで使用可能なプログラムが記録された媒体としても実施できる。したがって、本発明は、ハードウェアとしての実施形態、ソフトウェアとしての実施形態またはソフトウェアとハードウェアとの組合せの実施形態をとることができる。プログラムが記録された媒体としては、ハードディスク、ＣＤ－ＲＯＭ、光記憶装置または磁気記憶装置を含む任意のコンピュータ可読媒体を例示できる。

【００１７】

本発明の文書処理方法は、一般的なコンピュータシステムを用いて実現できる。本発明のシステムは、スタンドアロンのコンピュータシステムまたは複数のコンピュータシステムで構成されたコンピュータネットワークで実現できる。図１（ａ）はスタンドアロンのコンピュータを構成の概略を示し、図１（ｂ）は、コンピュータネットワークの場合を示す。

【００１８】

コンピュータシステムには、中央演算処理装置１（ＣＰＵ）、主記憶装置２（メインメモリ：ＲＡＭ）、不揮発性記憶装置３（ＲＯＭ）等を有し、バス４で相互に接続される。バス４には、その他コプロセッサ、画像アクセラレータ、キャッシュメモリ、入出力制御装置（Ｉ／Ｏ）等が接続されてもよい。また、バス４には、適当なインターフェイスを介して外部記憶装置５、データ入力デバイス６、表示デバイス７、通信制御装置８等が接続される。その他、一般的にコンピュータシステムに備えられるハードウェア資源を備えることが可能なことは言うまでもない。

【００１９】

外部記憶装置５は代表的にはハードディスク装置が例示できるが、これに限られず、光磁気記憶装置、光記憶装置、フラッシュメモリ等半導体記憶装置も含まれる。なお、データの読み出しのみに利用できるＣＤ－ＲＯＭ等の読み出し専用

記憶装置もデータあるいはプログラムの読み出しに適用する場合には外部記憶装置に含まれる。

【００２０】

データ入力デバイス６には、キーボード等の入力装置、マウス９等ポインティングデバイスを備えることができる。データ入力デバイスにはスキャナ等の画像読み取り装置、音声入力装置も含む。表示装置７としては、ＣＲＴ、液晶表示装置、プラズマ表示装置が例示できる。

【００２１】

複数のコンピュータシステムで本発明を実現する場合、図１（ｂ）に示すように、各コンピュータシステムは、ＬＡＮ、ＷＡＮ等で接続されていてもよく、また、インターネットを介して接続されても良い。これら接続に用いられる通信回線は、専用線、公衆回線の何れでも良い。コンピュータシステムには、パーソナルコンピュータ１０、ワークステーション１１、メインフレームコンピュータ１２等各種のコンピュータが含まれる。

【００２２】

コンピュータシステムが複数接続されたコンピュータネットワークにおいては、一部のプログラムをユーザのコンピュータで、一部のプログラムをリモートコンピュータで分散的に処理を実行できる。また、プログラムで利用されるデータは、それがどのコンピュータに記録されているかは問われない。つまり、データの所在に関する情報（アドレス）が明らかである限り、データあるいはプログラムの格納場所はコンピュータネットワーク上の任意の場所とすることができる。各ネットワークコンピュータ間の通信には公知の通信技術を適用でき、たとえばＴＣＰ／ＩＰ、ＨＴＴＰ等のプロトコルを用いることができる。また、各記憶装置に記録された各ファイル（データあるいはプログラム）の存在箇所（アドレス）は、ＤＮＳ、ＵＲＬ等を用いて特定できる。なお、本明細書においてインターネットという用語には、イントラネットおよびエクストラネットも含むものとする。インターネットへのアクセスという場合、イントラネットやエクストラネットへのアクセスをも意味する。コンピュータネットワークという用語には、公的にアクセス可能なコンピュータネットワークと私的なアクセスしか許可されない

コンピュータネットワークとの両方が含まれるものとする。

【００２３】

次に、本明細書で用いる用語を説明する。特に言及した場合を除き、本明細書では以下の概念で用語を用いる。

【００２４】

「オブジェクト」は、以下に説明するキャラクタ、スペース、トークン、テキストブロック等文書を構成する要素を総称する。

【００２５】

「シンボル」とは、空白文字を含むキャラクタのセットであり、「キャラクタ」とは、a,b,c等のアルファベット文字、記号等の独立したシンボルセットをいう。図３に示す網掛けの部分２１の「N」「S」「H」「R」「R」はキャラクタの例である。なお、漢字等の２バイト文字もキャラクタに含む。

【００２６】

「スペース」とは１行内の空白文字あるいはその連続したものをいう。図３に示す網掛けの部分２２はスペースの例である。２バイトの空白文字も含むが、１バイト空白文字の２文字分の連続と等価である。

【００２７】

「トークン」とは同一行内のキャラクタまたはその連続したものをいう。図３に示す網掛けの部分２３の「Exercise」はトークンの例である。

【００２８】

「テキストブロック」とはトークンのセットである。テキストブロックはトークンが含まれる最小面積の方形で表され、左上及び右下の座標で文書中の位置が記述できる。図４に示す網掛け部分２４はテキストブロックの例であり、９個のトークン「Number」「of」「Securities」「Underlying」「Options」「Granted」「(」「#」「)」が含まれる。なお、テキストブロックにはスペースが含まれてもよい。

【００２９】

後に説明するようにキャラクタおよびスペースはトークンの生成に用いられ、トークンはテキストブロックの生成に用いられる。トークン、スペースおよびテ

キストブロックは、その位置座標と共にデータベースに記録され蓄積される。このように位置座標と共にトークン、スペースおよびテキストブロック（オブジェクト）をデータベースに記録するため、これらオブジェクトの文書における実際の位置検索が速やかにできるようになる。

【００３０】

また、本明細書ではオブジェクトの抽象化のために「グラフ」、「グラフセット」、「単一要素サブクラスタ」および「複雑度」の概念を用いる。

【００３１】

「グラフ」とはノード（点）と弧（辺）のセットである。図５(a)にグラフの一例を示す。ノード２５間は方向を持つ弧２６で接続される。弧２６の始点はソースであり弧の終点はシンクである。

【００３２】

「グラフセット」とは、グラフのセットである。図５(b)にグラフセットの一例を示す。

【００３３】

「単一要素サブクラスタ」とは、グラフの部分グラフであり、各々のノードから出る弧の数（出次数）およびノードに入る弧の数（入次数）が１のグラフである。図６に単一要素サブクラスタの一例を示す。矢印の左側に示すグラフから単一要素サブクラスタを抽出したものが矢印の右側に示されている。２つのノード２７，２８については入次数が１であるが、そのソースであるノード２９の出次数が２であるからノード２７，２８が除外されて単一要素サブクラスタが構成される。

【００３４】

グラフセットあるいはグラフを構成するノードに関連するリンク（ノードに入出する弧数）の度合いは複雑度で表される。「複雑度」とは、ソース（ノード）から出るリンク（弧）の数とシンク（ノード）に入るリンク（弧）の数の和である。従って、単一要素サブクラスタの複雑度は２となる。また、あるグラフセットにおける最大複雑度は、グラフセット内の全ての弧における最大複雑度である。

【００３５】

本発明では、これらグラフの概念を用いて文書を表現する。各ノードがテキストブロックに対応し、弧がテキストブロック間のリンク（接続関係）に対応する。シンク（弧の終点となるノード）はソース（弧の始点となるノード）からの接続可能性のあるテキストブロックを表す。単一のテキストブロックから複数の弧が出ている場合には、複数のテキストブロックへの接続可能性を有することになる。

【００３６】

また、本発明ではテキストブロックと同様にスペースをグループ化してストリームを生成する。「ストリーム」とは、文書内の各行において上下に位置する各ノードを相互に接続したスペース（ノード）で構成されるグラフである。ストリームの長さは上下に延びる空白行の行数で表される。図７にストリームの一例を示す。図示するようにスペース３０の上下の広がりにより長さ５のストリームが構成されている。

【００３７】

以下、本実施の形態の文書処理方法を説明する。まず、前記したようなシステムに処理対象となる文書を入力する。入力はたとえばスキャナ等の読み取り装置で入力されるほか、既に電子化された文書データとして入力される。ただし、電子化された文書であっても、空白文字、タブ等でレイアウトされている文書であれば十分であり、高度に構造化されている必要はない。図２は、本システムに入力される文書の一例を示す図である。図２に示す文書は文字等のキャラクタで構成された単一ファイルである。ここで、文書とは、一対の座標で各々独立に特定される文字、空白等シンボルの集合と定義できる。図２において、左上の位置を（０，０）とし、横方向（ｘ方向）に１文字づつｘ座標指標が増加し、下方向（ｙ方向）に１行ずつｙ指標が増加するように座標を定義付ける。たとえば上から６行目の左側に表示されているテキスト「Name」の「N」の座標は（２，５）である。また、文書の行数はｙの最大値maxyであり、行内におけるシンボルの数はｘの最大値maxxである。このように座標に関連付けて１文字ずつデータベースに記録する。なお、次に説明するトークン生成処理と連動して、各行ごとに行の初

めから１文字ずつ右方向にシンボルを読み取る方式により入力されてもよい。

【００３８】

図８は本実施の形態の処理の概要を示したフローチャートである。文書データを入力後、初期化処理を行い（ステップ３１）、次に単一要素サブクラスタの結合処理を行う（ステップ３２）。その後、ユニークなリンクを有するクラスタ間を結合し（ステップ３３）、最後に残ったクラスタのリンクを評価してテキストブロックを生成する（ステップ３４）。

【００３９】

初期化処理を説明する。初期化処理は、４つのステップで行われる。第１のステップはトークン生成ステップである。第２のステップはストリーム生成ステップである。第３のステップは初期テキストブロックの生成ステップであり、第４のステップは初期リンクの生成ステップである。

【００４０】

図９は、トークン生成ステップの一例を示したフローチャートである。ステップ４０から処理を開始する。文書データの列方向の指標ｙに０を代入して初期化し（ステップ４１）、ｙが最大行数ｍａｘｙより小さいかを判断する（ステップ４２）。ステップ４２の判断がｎｏなら処理を終了し（ステップ４３）、それ以外は以下の処理を行う。なお、図中「＝」の記号は代入記号であり、以下同様である。

【００４１】

文書データの行方向の指標ｘに０を代入してｘを初期化し（ステップ４４）、変数startにｘを代入する（ステップ４５）。ｘが最大文字数ｍａｘｘを超えないかを判断し（ステップ４６）、ステップ４６の判断がｎｏならｙを１つ増加して（ステップ４７）ステップ４２に戻り、次行の処理に進む。それ以外の場合には以下の処理（ｙ行内のキャラクタのトークン化）を行う。

【００４２】

まず、変数Ｔに関数char_type(x,y)の戻り値を代入する（ステップ４８）。関数char_type(x,y)は、座標（ｘ、ｙ）の位置にあるシンボルのキャラクタタイプを戻り値として返す関数である。本実施の形態では、アルファベット、数字、句

読点、スペースをキャラクタタイプとして考慮する。ただし、日本語等英語以外の言語を処理する時には漢字等他の文字を考慮してもよいことは勿論である。

【００４３】

次に、char_type(x,y)の戻り値と変数Ｔの値が等しいかを判断する（ステップ４９）。なお、図中「＝＝」の記号は両辺の値が等しいか否かを判断する記号であり、以下同様である。最初のループでは前記判断は「真（ｙｅｓ）」になるのでステップ５０に進みｘを１つ増加する。ｘがｍａｘｘ以下であることを判断し（ステップ５１）、ｙｅｓであればステップ４９に戻る。ｘが１つ増加するのでステップ４９ではｙ行内の隣接するシンボルのタイプを検査することになる。シンボルタイプが同じ（ステップ４９の判断がｙｅｓ）の場合にはｘがｍａｘｘを超えない範囲でステップ５０、５１のループを繰り返し、タイプの異なるシンボルが検出されるまでこのループが繰り返される。異なるタイプのシンボルが検出されると（ステップ４９の判断がｎｏ）ステップ５２に進み、変数tokenにこれら同一タイプの連続するシンボルの座標が入力される(ステップ５２）。なお、ステップ５１でｎｏと判断された場合（行末まで処理が進んだ場合）にもステップ５２に進む（ステップ５１）。

【００４４】

次にキャラクタタイプがスペースであるかを判断する（ステップ５３）。スペースであるときには変数tokenをスペースデータベースに追記し（ステップ５４）、スペースでない場合には変数tokenをトークンデータベースに追記する（ステップ５５）。その後ステップ４５に進み、前記処理を繰り返す。

【００４５】

このようにして入力文書のトークン化処理が行われる。なお、前記の通りトークン化と同時にスペースの検出も行われる。図１０は、トークン化処理が終了した後の文書の一例を示す。たとえば０行目に着目すれば、ｘが０～２５の範囲でスペースが検出され、１つのスペース（token）としてスペースデータベースに記録される。ｘが２６～３５の範囲の「ｉ」「ｎ」「ｄ」「ｉ」「ｖ」「ｉ」「ｄ」「ｕ」「ａ」「ｌ」が同一タイプのキャラクタなのでトークン「ｉｎｄｉｖｉｄｕａｌ」が生成され、トークンデータベースに記録される。なお、トークン

生成の手法としてchar_type関数を用いる例を示したが、その他個別キャラクタからトークンを発生させる方法は種種存在し、上記の手法には限られない。

【００４６】

次に、ストリーム生成手法を説明する。ストリームはスペースデータベースを用いて計算される。図１１は、ストリームサイズの計算方法の一例を示したフローチャートである。文書データの列方向の指標ｙに０を代入して初期化し（ステップ５６）、ｙが最大行数ｍａｘｙより小さいかを判断する（ステップ５７）。ステップ５７の判断がｎｏなら処理を終了し（ステップ５８）、それ以外は以下の処理を行う。

【００４７】

ｙ行に存在するスペースを変数Ｓに代入し（ステップ５９）、変数ｎにＳの数｜Ｓ｜を代入する（ステップ６０）。なお、｜Ｏ｜はオブジェクトＯの数を示すスカラー値であり、以下同様である。また、変数Ｓはベクトル量であり、ベクトルの各要素にスペース（オブジェクト）が代入される。以下変数Ａ、変数Ｔにおいて同様である。

【００４８】

変数ｉに０を代入して初期化し（ステップ６１）、ｉがｎより小さいかを判断する（ステップ６２）。ステップ６２の判断がｙｅｓなら変数spaceにｉ番目のスペースＳ［ｉ］を代入し（ステップ６３）、変数Ａにｙ－１行目におけるスペースのうち、スペースＳ［ｉ］のｘ方向位置が一致するスペースを代入する(ステップ６４）。そして変数space.aboveに、スペースＡのうち何れかのスペースs'の持つ上部スペース数の最大値（max s'.avobe）に１を加えた数を代入する（ステップ６５）。ここで、変数s.aboveには、スペースsの上部にあるスペース数が記録されている。つまり前記処理により、スペースspaceの上部に存在するスペース数として、スペースspaceにｘ方向位置が一致するベクトル量Ａの要素s'のうち最大の上部スペース数（max s'.avobe）に１を加えた数が代入される。space.aboveは、spaceの上部に存在する連続したスペースの数（行数）を示す。

【００４９】

その後、ｉに１を加えて（ステップ６６）、ステップ６２に戻る。このように

してｙ行目に存在する各スペース（Ｓ）の上部にあるスペース数が計算される。この操作をｍａｘｙまで繰り返す（ステップ６７）。

【００５０】

上記手段により任意のスペース上にあるスペース数が計算でき、所定の閾値を超えた時にはこれをストリームと判断してストリームデータベースに記録できる。図１２は、ストリームを計算した後の結果を示す文書である。網掛けブロックで示したスペース６８がストリームを構成する。

【００５１】

次に、初期テキストブロックの生成方法を説明する。初期テキストブロックは、トークンデータベース、スペースデータベースおよびストリームデータベースを用いて計算される。図１３は初期テキストブロックの計算方法の一例を示したフローチャートである。文書データの列方向の指標ｙに０を代入して初期化し（ステップ６９）、ｙが最大行数ｍａｘｙより小さいかを判断する（ステップ７０）。ステップ７０の判断がｎｏなら処理を終了し（ステップ７１）、それ以外は以下の処理を行う。

【００５２】

ｙ行に存在するトークンを変数Ｔに代入し（ステップ７２）、ｙ行に存在するスペースを変数Ｓに代入する（ステップ７３）。そして変数ｎにＴの数｜Ｔ｜を代入する（ステップ７４）。前記した通り、変数Ｔはベクトル量であり、ベクトルの各要素にトークン（オブジェクト）が代入される。

【００５３】

変数ｉに０を代入して初期化し（ステップ７５）、ｉがｎより小さいかを判断する（ステップ７６）。ステップ７６の判断がｙｅｓならｉがｎ－１であるかを判断する（ステップ７７）。つまり現在のｉ番目のトークンがｙ行における最後のトークンであるかを判断する。このステップ７７の判断がｎｏであるときには変数ｔにｉ番目のトークンＴ［ｉ］を代入する(ステップ７８）。その後変数ｓにトークンｔの右側に位置するスペースを代入する（ステップ７９）。そして、スペースｓ（トークンｔの右側に位置する）がストリームに属するかを判断する（ステップ８０）。s.stream関数はスペースｓがストリームに属する時には新値

を戻す関数である。

【００５４】

ステップ８０の判断がｙｅｓ（トークンｔの右側のスペースがストリームである）の場合には、トークンｔをテキストブロックデータベースに追加する（ステップ８１）。なお、ステップ７７でｙｅｓ（トークンがｙ行における最後のトークンである）と判断された時にはステップ８１に進む。

【００５５】

一方ステップ８０の判断がｎｏである時にはスペースｓの大きさ｜ｓ｜があらかじめ定めた最大スペース値（maxspace）より大きいかを判断し（ステップ８２）、ステップ８２の判断がｙｅｓの時にはステップ８１に進んでトークンｔをテキストブロックデータベースに追加する。ステップ８２の判断がｎｏの時にはトークンｔ’としてｉ＋１番目のトークンＴ［ｉ＋１］を代入し、さらにトークンｔとｔ’とをマージしてトークンｔとする。さらにトークンＴ［ｉ］にトークンｔを代入し、スペース列Ｓからスペースｓを削除し、ｎから１を減じる（ステップ８３）。つまり、ストリームでない空白の両側に位置するトークンをマージする処理を行う。その後ステップ７６に進んで上記処理を繰り返す。

【００５６】

なお、テキストブロックデータベースにトークンが記録された後は、ｉを１増加し（ステップ８４）、ステップに進んで上記処理を繰り返す。

【００５７】

そしてステップ７６でｎｏと判断された時（１行分のトークンのマージ処理が終了した時）にはｙを１増加し（ステップ８５）、ステップ７０に戻って処理を繰り返す。

【００５８】

すなわち、上記処理により、ストリームまたは行の終端が検出されるまでは１行内のトークンはマージされる。このマージされたトークンが初期テキストブロックとして、テキストブロックデータベースに記録される。

【００５９】

図１４は、初期的テキストブロックが生成された段階の文書の例を示す図であ

る。前記処理フローより明らかな通り、この段階でのテキストブロックは１行内でのトークンのマージに止まるため、その深さは１である。また、同図に示すように、たとえば「Employees」と「in」との間の領域８６のようにストリームの一部であるスペースが初期テキストブロックの間に残る。また、たとえば領域８７のように、ストリームでないスペースによってもトークンがマージされない場合がある。つまり、ステップ８２の判断において最大スペース値（maxspace）を越えた場合である。

【００６０】

次に、初期リンクの生成を行う。初期リンクはの生成は、テキストブロックデータベースとスペースデータベースを用いて計算する。図１５は初期リンク生成の一例を示したフローチャートである。文書データの列方向の指標ｙに０を代入して初期化し（ステップ８８）、ｙが最大行数ｍａｘｙより小さいかを判断する（ステップ８９）。ステップ８９の判断がｎｏなら処理を終了し（ステップ９０）、ｙｅｓなら以下の処理を行う。

【００６１】

ｙ行に存在するテキストブロックを変数Ｔに代入し（ステップ９１）、変数nextにｙ＋１を代入する（ステップ９２）。次にnext行が空であるかを判断し（ステップ９３）、空である場合には変数nextをさらに１つ増加し（ステップ９４）、空でない場合にはそのまま次のステップ９５に進む。ステップ９５ではnext行に存在するテキストブロックを変数Ｔ’に代入する。そして変数ｎにＴの数｜Ｔ｜を代入する（ステップ９６）。

【００６２】

変数ｉに０を代入して初期化し（ステップ９７）、ｉがｎより小さいかを判断する（ステップ９８）。ステップ９８の判断がｙｅｓなら、変数ｔにｉ番目のテキストブロックＴ［ｉ］を代入した後（ステップ９９）、ｉがｎ－１であるかを判断する（ステップ１００）。つまり現在のｉ番目のテキストブロックがｙ行における最後のテキストブロックであるかを判断する。このステップ１００の判断がｎｏであるときには、変数ｔ’にｉ＋１番目のテキストブロックＴ［ｉ＋１］を代入し（ステップ１０１）、ｔとｔ’との間のスペースをｓに代入する（ステ

ップ１０２）。ｓがストリームであるか（s.stream==true）もしくはｓの大きさ｜ｓ｜が最大リンクスペース（max link space）を超えているかを判断し（ステップ１０３）、ステップ１０３の判断がｎｏの時にはｔとｔ’との間にリンクを生成する（ステップ１０４）。つまりステップ９９からステップ１０４までの処理により、同一行内の隣接するテキストブロック間にストリームでないスペースが存在しかつそのスペースの大きさ（長さ）が最大リンクスペースより小さい時には隣接テキストブロック間にリンクを形成する。

【００６３】

次に変数Ｌにｔよりも左側に存在するＴ’内の全てのテキストブロックを代入する（ステップ１０５）。そして、ｔとＬを構成する各々のテキストブロック間にリンクを生成する（ステップ１０６）。つまり、着目しているテキストブロックｔの次行（次行が空行の時にはその次の行）に存在し、ｔよりも左側に位置するテキストブロックの全てにリンクを形成する。なお、ステップ１００、１０３においてその判断がｙｅｓ（着目テキストブロックｔに右隣接するテキストブロックがない、あるいはストリームまたはスペースが大きくてリンクを張るのが妥当でない）と判断された時にはステップ１０５に進む。

【００６４】

上記のようにリンクを形成した後ｉを１増加し（ステップ１０７）、ステップ９８に進んで上記処理を繰り返す。そしてステップ９８でｎｏと判断された時（１行分のテキストブロックについてリンク形成処理が終了した時）にはｙを１増加し（ステップ１０８）、ステップ８９に戻って処理を繰り返す。

【００６５】

すなわち、上記処理により、初期テキストブロック間のリンクが形成される。このリンク生成の判断基準は、隣接テキストブロック間のスペースがストリームでないこと、スペースが予め定めた最大リンクスペースを越えないこと、リンクの終点となるテキストブロック（シンク）の位置が、リンクの始点となるテキストブロック（ソース）の次行左側に位置することである。このような条件を満たせば自動的に初期リンクが生成される。初期リンクはリンクのソース、シンクとなるテキストブロックの情報とともにデータベースに記録されるのは勿論である

。

【００６６】

以上のようにして初期化処理が終了する。図１６は初期化処理のステップを擬似コードで表した図である。なお、擬似コードを表す図において左側に示した数字は行番号であり、以下同様である。

【００６７】

文書データdocをトークン化関数tokenizeに入力し、トークンtokensおよびスペースspacesを得る（行番号１）。また、文書データdocをストリーム関数steamに入力し、ストリームstreamsを得る（行番号２）。tokens、spaces、streamsを初期テキストブロック生成関数get_initial_blocksに入力し、初期テキストブロックとしてtext_blocksを得る（行番号３）。text_blocksを初期リンク生成関数get_initial_linksに入力し、初期リンクとしてlinksを得る（行番号４）。そして、初期テキストブロックtext_blocksおよび初期リンクlinksを文書グラフdocument_graphとしてストアする（行番号５）。文書グラフは、text_blocksをノード、linksを弧とするグラフセットである。

【００６８】

次に、単一要素サブクラスタの結合処理（ステップ３２）を説明する。図１７は単一要素サブクラスタ結合処理の一例を示した擬似コードを示す図であり、図１８はそのフローチャートである。

【００６９】

まず、cluster関数を用いて文書をクラスタ化する（行番号１）。クラスタ化されたデータはcluster_setに格納される。文書のクラスタ化は前記した初期化処理で生成したテキストブロックとリンクで表現される文書からグラフを取り出すことにより行う。リンクで結合されているテキストブロックの集合が１つのグラフに対応する。

【００７０】

次にcluster_setに含まれるクラスタｃの全てについてsub-cluster関数を用いてサブクラスタ化する。抽出されたサブクラスタはsub_cluster_setに格納される（行番号２，３、ステップ１０９，１１０）。サブクラスタ化は、クラスタか

ら単一要素のサブクラスタを抽出する作業である。たとえば各ノードにおける弧（リンク）の入次数および出次数が１となる条件（単一要素サブクラスタの定義）を満足するかをチェックしながら抽出できる。

【００７１】

次にsub_cluster_set に含まれるサブクラスタｓの全てについて、各サブクラスタｓに含まれるリンクの妥当性を言語モデルを用いて評価する（行番号４～６、ステップ１１１～１１３）。リンクの妥当性は、リンクの始点（ソース）のテキストブロックとリンクの終点（シンク）のテキストブロックとが、言語モデルにおいて高い確率で出現する表現であるかを評価することにより行う。言語モデルにはたとえばNグラムモデルを用いることができる。ただしNグラムモデルに限らず、その他のモデルであっても構わない。評価が妥当である場合にはソースとシンクのテキストブロックがマージされる（号番号６、ステップ１１４）。

【００７２】

次に、クラスタ結合処理（ステップ３３）を説明する。図１９はクラスタ結合処理の一例を示した擬似コードを示す図であり、図２０はそのフローチャートである。クラスタの結合処理は複雑度の小さなものから順に行う。

【００７３】

まず、複雑度に３を代入しておき（行番号１）、最大複雑度を取得する（行番号２）。次に複雑度が最大複雑度よりも小さいときには以下の処理を行う（行番号３）。

【００７４】

cluster関数を用いて文書をクラスタ化し（行番号５）、cluster_setに含まれるクラスタｃの全てについて、各クラスタｃに含まれるリンクの複雑度をチェックする（行番号６～８、ステップ１１５～１１６）。リンクの複雑度が現ループの複雑度より小さければ（行番号８、ステップ１１６）、リンクがユニークに妥当であると評価できるかを判断する（行番号９、ステップ１１７）。判断の結果ｙｅｓであれがリンクのソースとなるテキストブロックとシンクとなるテキストブロックをマージする（行番号９、ステップ１１８）。なおリンクがユニークに妥当であるかどうかとは、唯一確かなリンクしか存在しないことをいう。たとえ

ば「number」「of」間のリンクと「of」「of」間のリンクが並存する場合、言語モデルからは「of」「of」間のリンクはありえない。この場合、「of」「of」間のリンクが取り去られ、「number」「of」間のリンクが選択される。そして「number」「of」がマージされて新たなテキストブロック「number of」が生成される。

【００７５】

次に、クラスタ間の接続評価ステップ（ステップ３４）を説明する。図２１はクラスタ間接続評価処理の一例を示した擬似コードを示す図であり、図２２はそのフローチャートである。この処理はクラスタの結合処理（ステップ３３）と類似している。ステップ３３では複数のリンクを評価した時に、唯一選択し得るリンクが存在する時にはこのリンクを選択する処理を行った。ここでは、複数のリンクが存在し、何れも選択可能な時の処理を説明する。なお、ステップ３３と同様な事項の説明は省略する。

【００７６】

ステップ３４では、cluster_setに含まれるクラスタｃの全てについて、各クラスタｃに含まれるリンクの順位付けを行う（行番号７、ステップ１１９）。順位付けられたリンクordered_linksに含まれる各リンクについて（行番号８、ステップ１２０）、リンクの複雑度をチェックし（行番号９、ステップ１２１）、リンクの妥当性評価に有意な差があるかを判断する（行番号１０、ステップ１２２）。リンクの妥当性評価には、たとえば言語モデルによる出現確率を用いることができる。出現確率が高いほど接続妥当性は高くなる。なおその他の妥当性評価および有意差の判定手法を用いることができることは勿論である。前記判断の結果ｙｅｓであれがリンクのソースとなるテキストブロックとシンクとなるテキストブロックをマージする（行番号１０、ステップ１２３）。なお、有意差が見られない時にはリンクは並存したままである。

【００７７】

このようにして空白でレイアウトされた文書から意味のあるテキストブロックを自動的に生成できる。図２３は最終的な処理後の出力の一例を示した図である。言語モデルにより妥当と判断された接続をマージして得られたテキストを含む

テキストブロックが表示されている。図示するように本実施の形態の処理によるテキストブロックは原文書の空間座標情報を維持したまま出力される。つまり最終的なテキストブロックの位置は、文書の位置座標で特定できる。たとえばテキストブロックの左上の座標と右下の座標で位置が特定される。このように原文書の空間情報が維持され、かつ、各テキストブロック内のテキストは言語モデルで保証された意味のある内容を含む。このため、本実施の形態の処理方法を前処理手段として高度な自然言語処理（たとえばテキストマイニングや機械翻訳）を原文書の内容を欠落することなく適用できる。さらに、本実施の形態の文書処理方法は、空間的な位置情報とスペースを利用してテキストブロックを生成するので、複雑なレイアウトを持つ文書にも容易に適用することが可能である。このため、適用の対象となる文書の範囲を広げ、より汎用的な文書処理に供することができる。

【００７８】

以上、本発明者によってなされた発明を発明の実施の形態に基づき具体的に説明したが、本発明は前記実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能である。

【００７９】

たとえば、入力文書には、前記実施の形態で用いた表に限らず、２段組等の複数段表示文書、リスト、表題が付された文書、ダブルスペース文書、マルチカラムセルあるいはマルチロウセルを有する表文書、サブセルを有する表文書、省略形のリストあるいは表等の文書にも適用できる。適用可能な文書の例を図２４の示す。

【００８０】

また、日本語等縦書きを含む文書にも適用できる。前記実施の形態では横書き文書を前提に説明したが、縦書き文書に適用するようにアルゴリズムを変更することは容易である。また、横書きと縦書きが混在するような文書にも適用できる。

【００８１】

さらに１つのテキストが複数のテキストにかかる場合や、複数のテキストが１

つのテキストにかかる場合のように文書間のかかり受けが複雑になる場合にも本実施の形態を用いれば正確に把握できる。この場合１つのオブジェクト（テキストブロック）に複数のリンクが入る（または出る）状態で記録される。

【００８２】

また、前記実施の形態では、単一要素のサブクラスタにおけるテキストブロック結合処理（ステップ３２）、クラスタ結合処理（ステップ３３）、およびクラスタ間接続評価（ステップ３４）の全てのステップを有する例を説明したが、ステップ３２あるいはステップ３４については必須のステップではない。入力文書によっては単一要素サブクラスタが存在しない場合があり、また、唯一のリンク可能性が発見できない場合もある。この場合、ステップ３４のテキストブロックの接続評価により、本実施の形態と同じ文書出力を得ることができる。ただし、ステップ３２、３３を入れることにより処理の効率化が図れる。

【００８３】

【発明の効果】

本願で開示される発明により、表、箇条書き、多段組等任意にレイアウトされた文書から意味のあるテキストブロックを抽出することができる。

【図面の簡単な説明】

【図１】

（ａ）はスタンドアロンのコンピュータを構成の概略を示し、（ｂ）は、コンピュータネットワークの場合を示す。

【図２】

本システムに入力される文書の一例を示す図である。

【図３】

キャラクタ、スペース、トークンの例を示す図である。

【図４】

テキストブロックの例を示す図である。

【図５】

(a)はグラフの一例を示し、(b)はにグラフセットの一例を示す図である。

【図６】

単一要素サブクラスタの一例を示す図である。

【図７】

ストリームの一例を示す図である。

【図８】

本発明の一実施の形態である処理の概要を示したフローチャートである。

【図９】

トークン生成ステップの一例を示したフローチャートである。

【図１０】

トークン化処理が終了した後の文書の一例を示す図である。

【図１１】

ストリームサイズの計算方法の一例を示したフローチャートである。

【図１２】

ストリームを計算した後の結果の一例を示す図である。

【図１３】

初期テキストブロックの計算方法の一例を示したフローチャートである。

【図１４】

初期的テキストブロックが生成された段階の文書の一例を示す図である。

【図１５】

初期リンク生成の一例を示したフローチャートである。

【図１６】

初期化処理のステップを擬似コードで表した図である。

【図１７】

単一要素サブクラスタ結合処理の一例を示した擬似コードを示す図である。

【図１８】

単一要素サブクラスタ結合処理の一例を示したフローチャートである。

【図１９】

クラスタ結合処理の一例を示した擬似コードを示す図である。

【図２０】

クラスタ結合処理の一例を示したフローチャートである。

【図２１】

クラスタ間接続評価処理の一例を示した擬似コードを示す図である。

【図２２】

クラスタ間接続評価処理の一例を示したフローチャートである。

【図２３】

最終的な処理後の出力の一例を示した図である。

【図２４】

入力文書のその他の例を示す表図である。

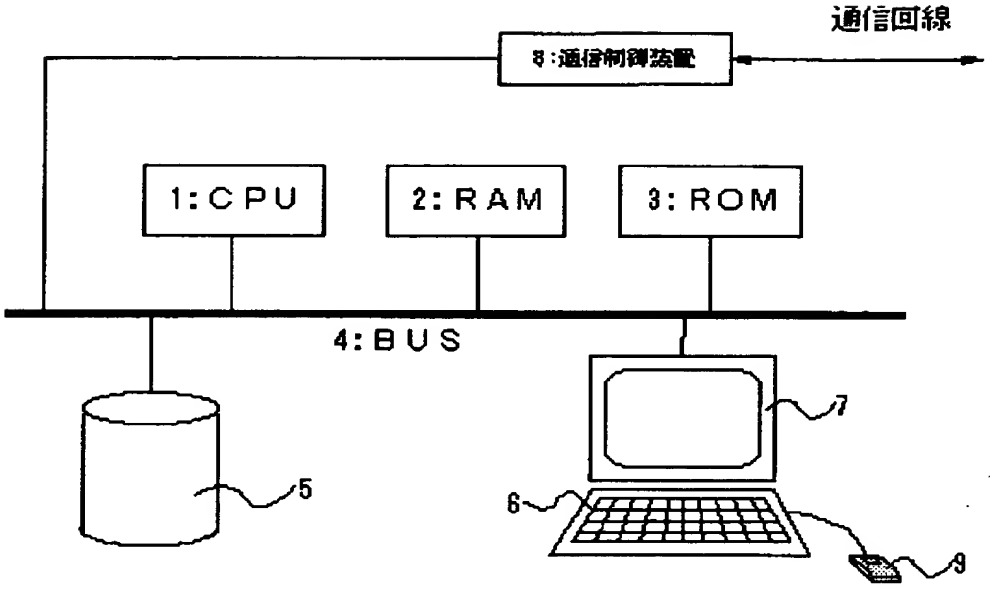
【符号の説明】

１…中央演算処理装置、２…主記憶装置、３…不揮発性記憶装置、４…バス、５…外部記憶装置、６…データ入力デバイス、７…表示デバイス（表示装置）、８…通信制御装置、９…マウス、１０…パーソナルコンピュータ、１１…ワークステーション、１２…メインフレームコンピュータ、２５，２７，２９…ノード、２６…弧、Ｔ，ｔ…テキストブロックまたはトークン、ｃ…クラスタ、ｍａｘｘ…最大文字数、ｍａｘｙ…最大行数、ｓ…サブクラスタまたはスペース。

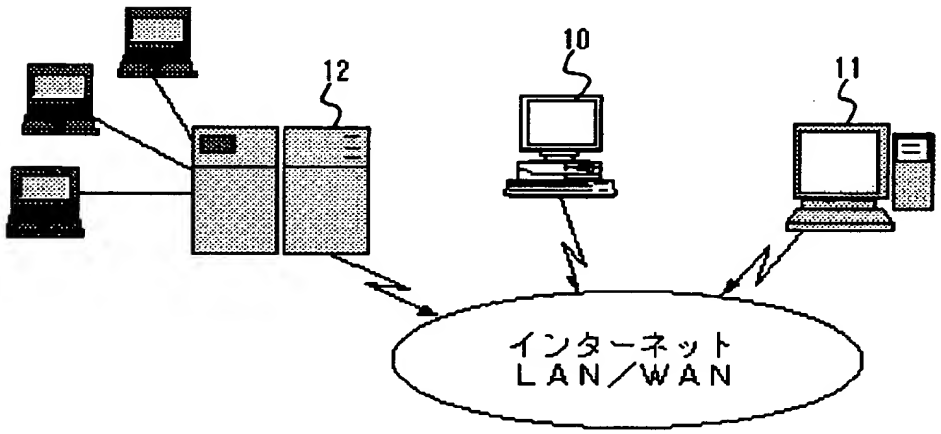
【書類名】　　図面

【図１】

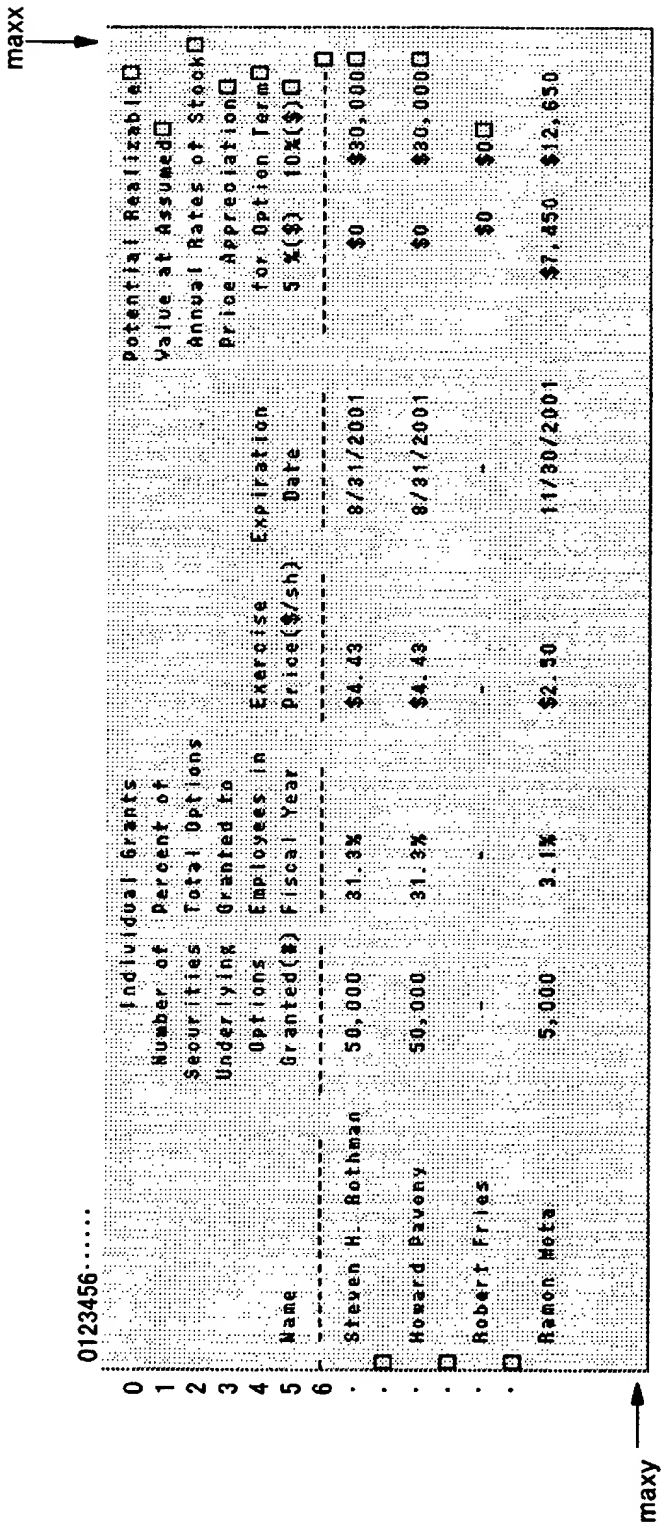
(a)


通信回線
8:通信制御装置
1:CPU
2:RAM
3:ROM
4:BUS
5
6
7
9

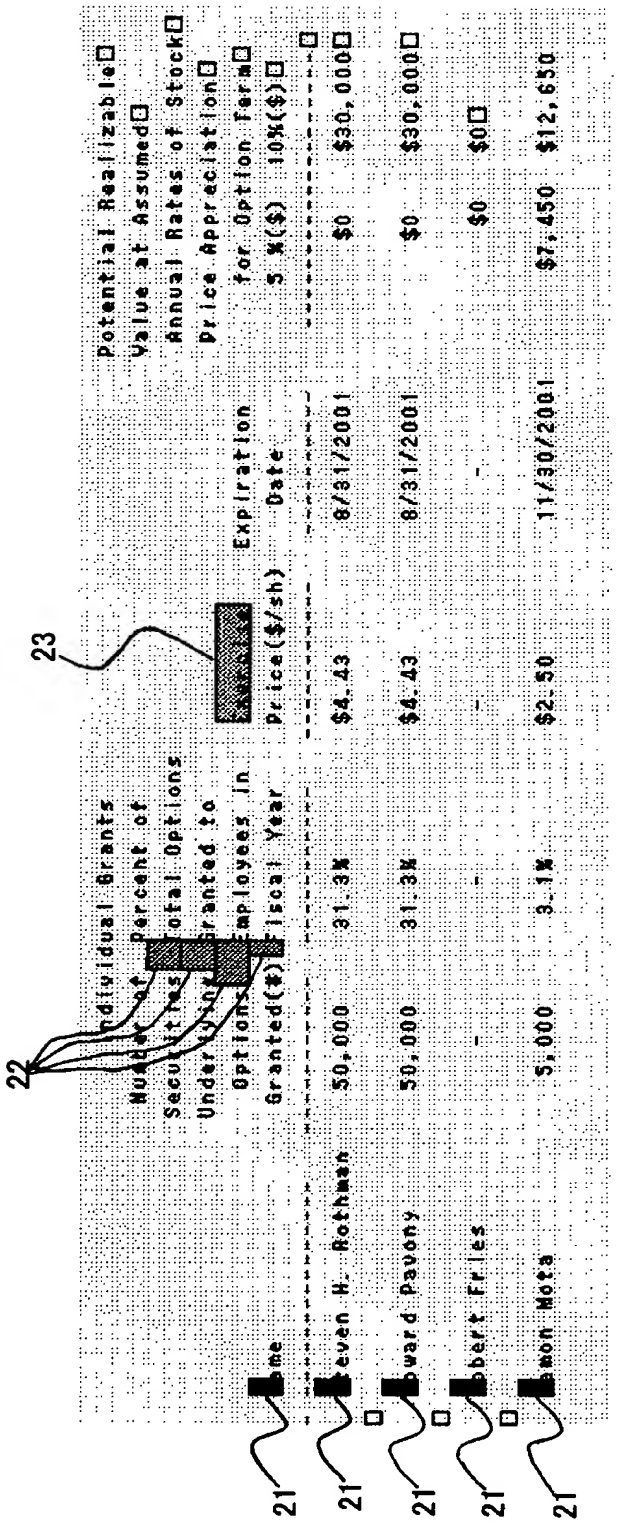
(b)


10
11
12
インターネット
LAN／WAN

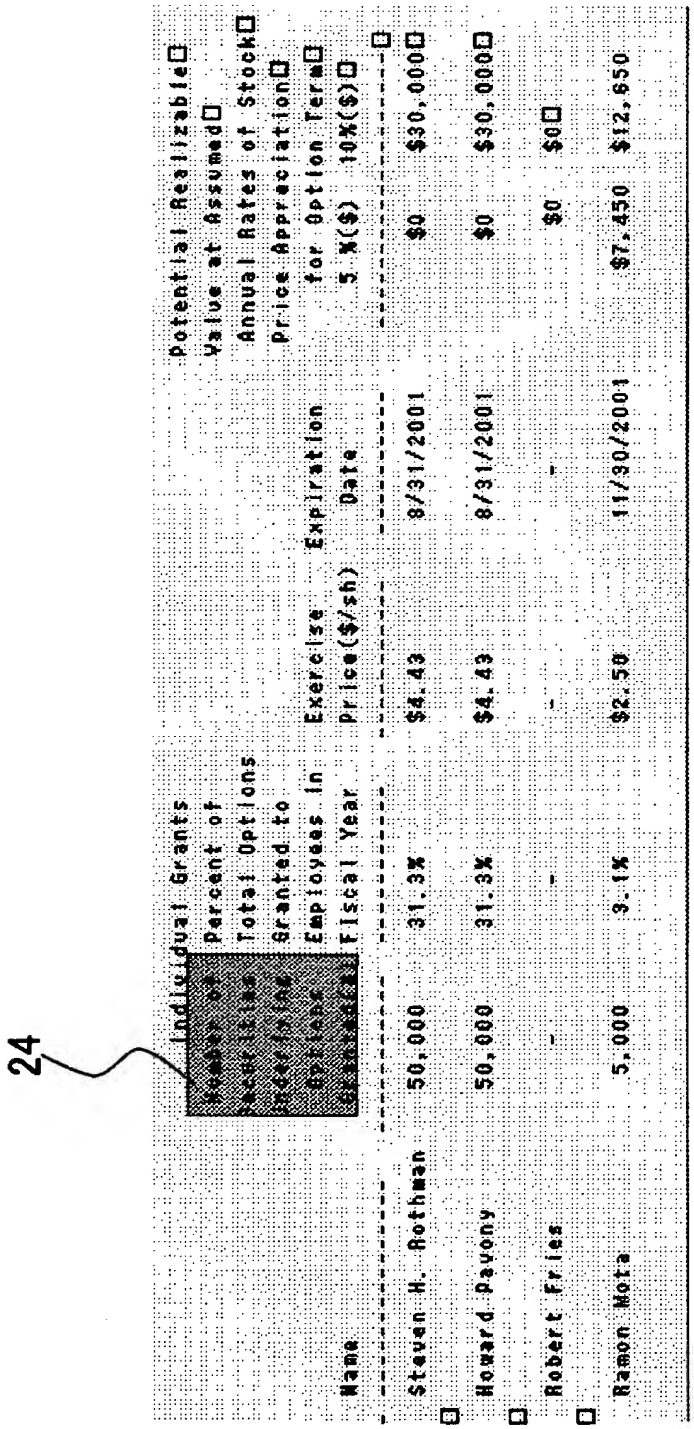
【図２】


maxx
maxy
0123456
Individual Grants
Name
Number of Securities Underlying Options Granted(#)
Percent of Total Options Granted to Employees in Fiscal Year
Exercise Price($/sh)
Expiration Date
Potential Realizable Value at Assumed Annual Rates of Stock Price Appreciation for Option Term
5 %($)
10%($)
Steven H. Rothman
50,000
31.3%
$4.43
8/31/2001
$0
$30,000
Howard Bavony
50,000
31.3%
$4.43
8/31/2001
$0
$80,000
Robert Fries
-
-
-
-
$0
$0
Ramon Mota
5,000
3.1%
$2.50
11/30/2001
$7,450
$12,650

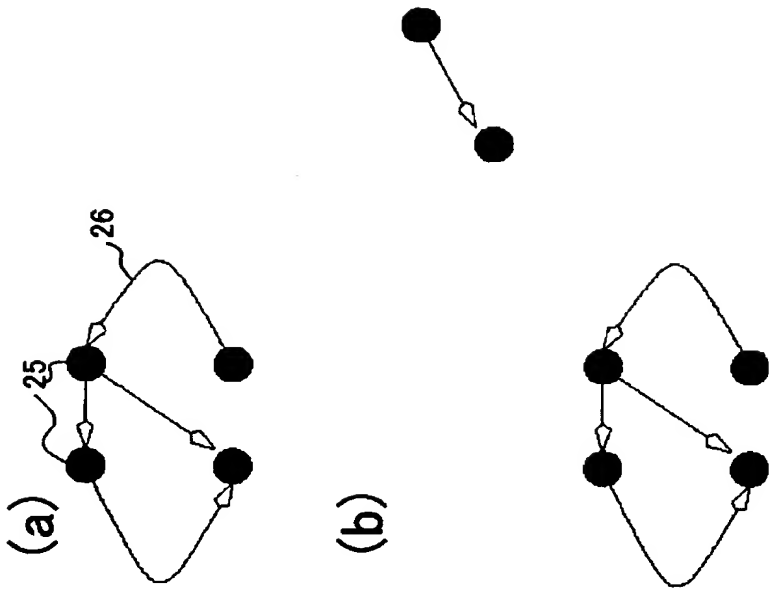
【図３】


Potential Realizable
Value at Assumed
Annual Rates of Stock
Price Appreciation
Individual Grants
Number of Percent of
Securities Total Options
Underlying Granted to
Options Employees in
Granted(#) Fiscal Year
Price($/sh)
Expiration Date
for Option Term
5 %($) 10%($)
50,000 31.3% $4.43 8/31/2001 $0 $30,000
50,000 31.3% $4.43 8/31/2001 $0 $30,000
- - - - $0 $0
5,000 3.1% $2.50 11/30/2001 $7,450 $12,650
me
even H. Rothman
ward Pavony
bert Fries
mon Mota
22
23
21
21
21
21
21

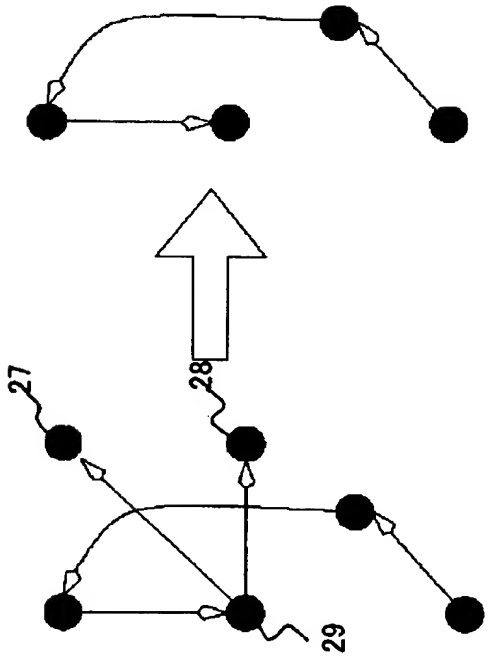
【図４】


24
Individual Grants
Potential Realizable
Number of Securities Underlying Options Granted to Employees in Fiscal Year
Percent of Total Options Granted to Employees in Fiscal Year
Value at Assumed Annual Rates of Stock Price Appreciation for Option Term
Exercise or Base Price($/sh)
Expiration Date
5 %($)
10%($)
Name
Steven H. Rothman
50,000
31.3%
$4.43
8/31/2001
$0
$30,000
Howard Pavony
50,000
31.3%
$4.43
8/31/2001
$0
$30,000
Robert Fries
-
-
-
-
$0
$0
Ramon Mota
5,000
3.1%
$2.50
11/30/2001
$7,450
$12,650

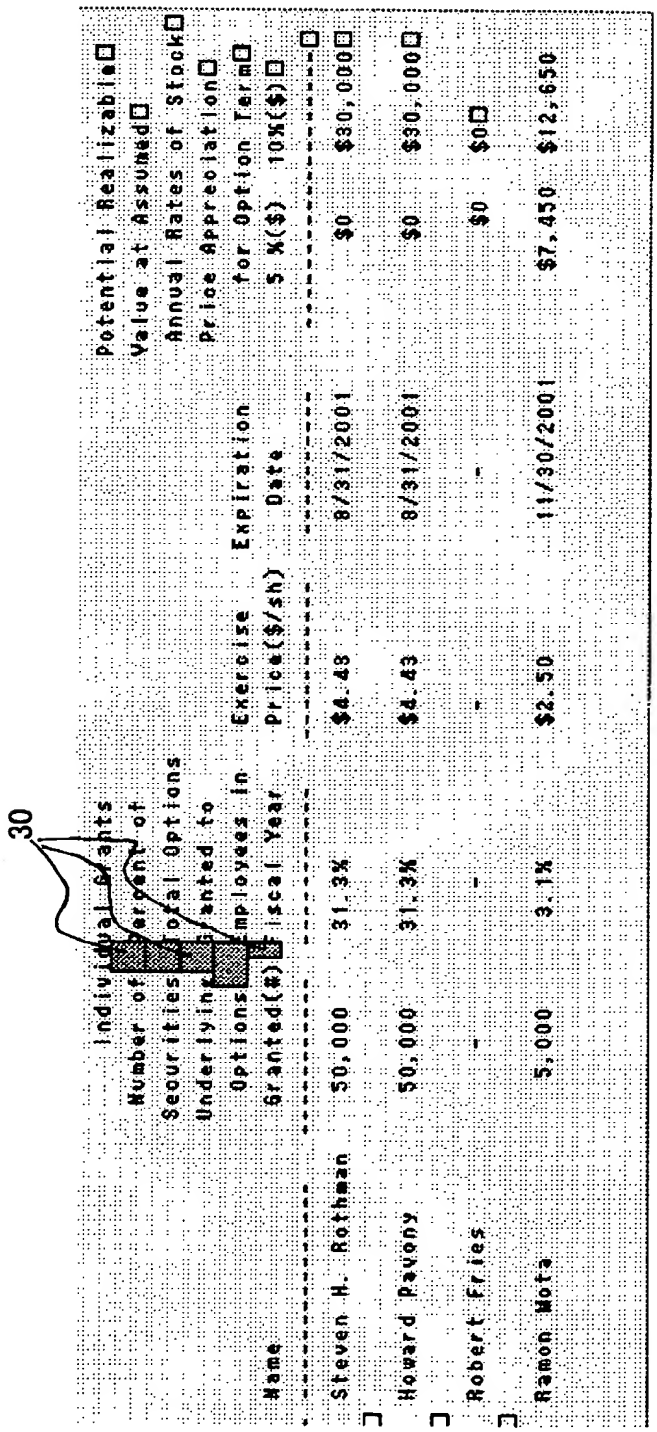
【図５】



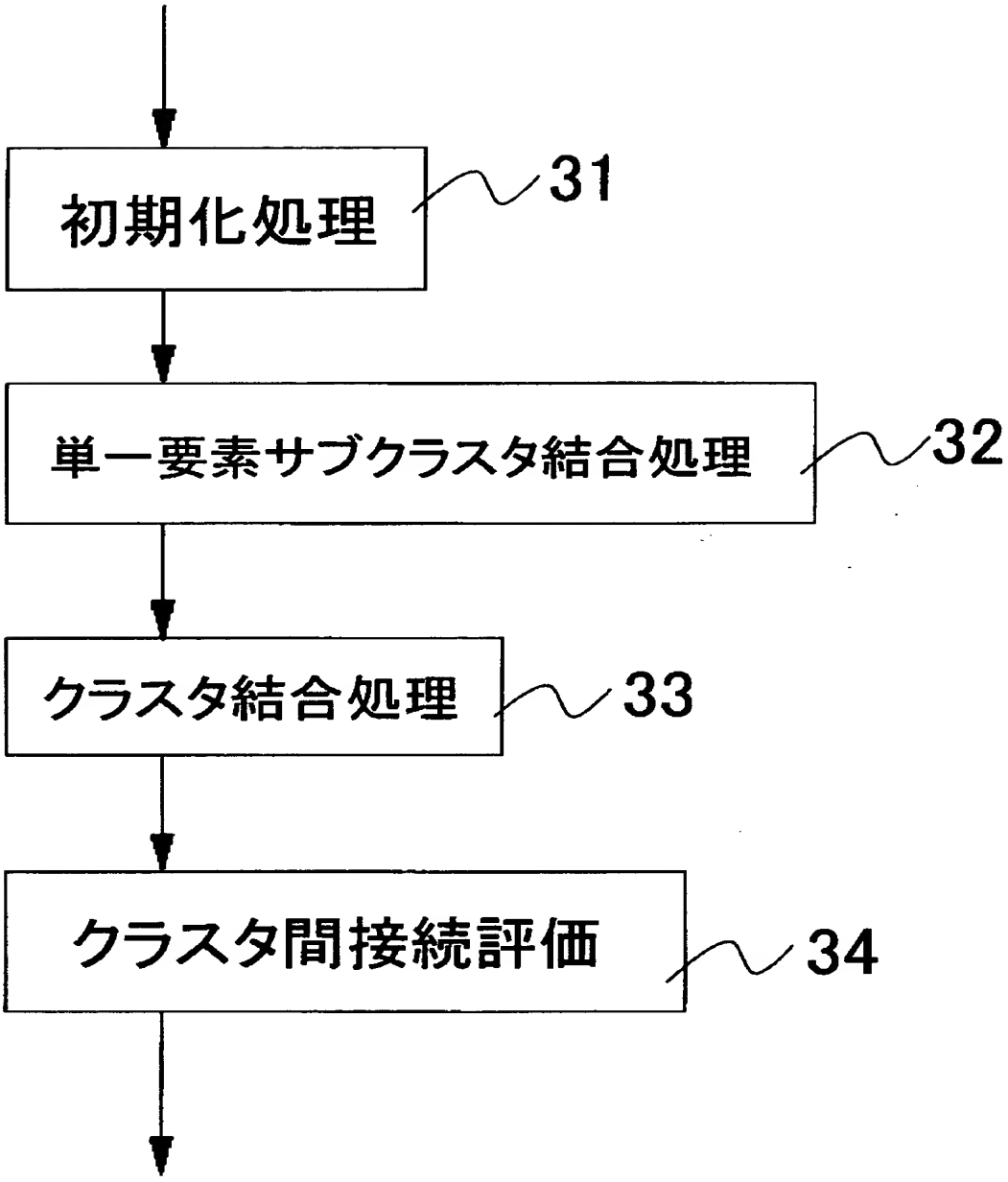
【図６】



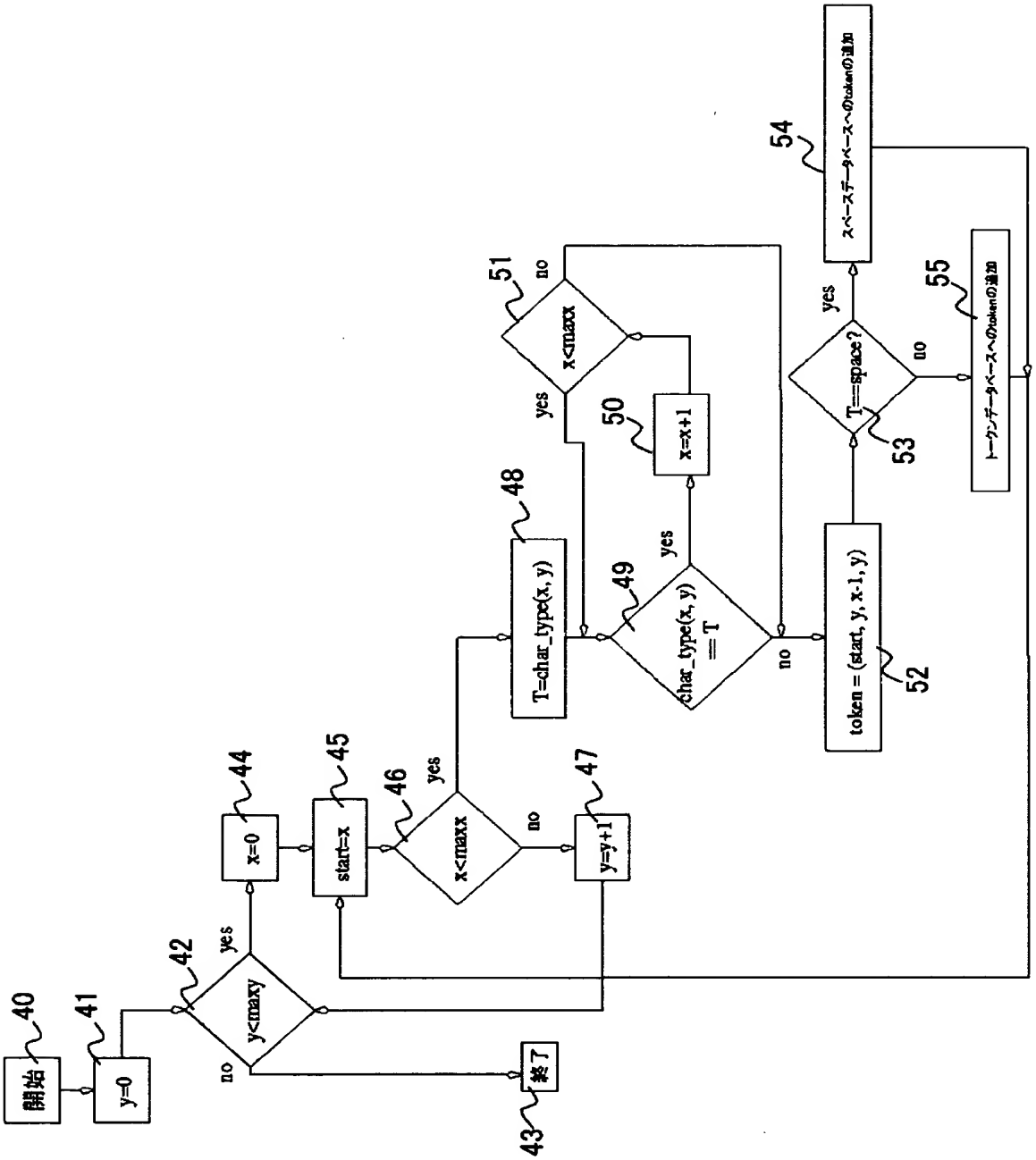
【図７】


30
Individual Grants
Number of Percent of Securities Total Options Underlying Granted to Options Employees in Fiscal Year
Exercise Price($/sh)
Expiration Date
Potential Realizable Value at Assumed Annual Rates of Stock Price Appreciation for Option Term
5 %($) 10%($)
Name
Steven H. Rothman 50,000 31.3% $4.43 8/31/2001 $0 $30,000
Howard Pavony 50,000 31.3% $4.43 8/31/2001 $0 $30,000
Robert Fries - - - - $0 $0
Ramon Mota 5,000 3.1% $2.50 11/30/2001 $7,450 $12,650

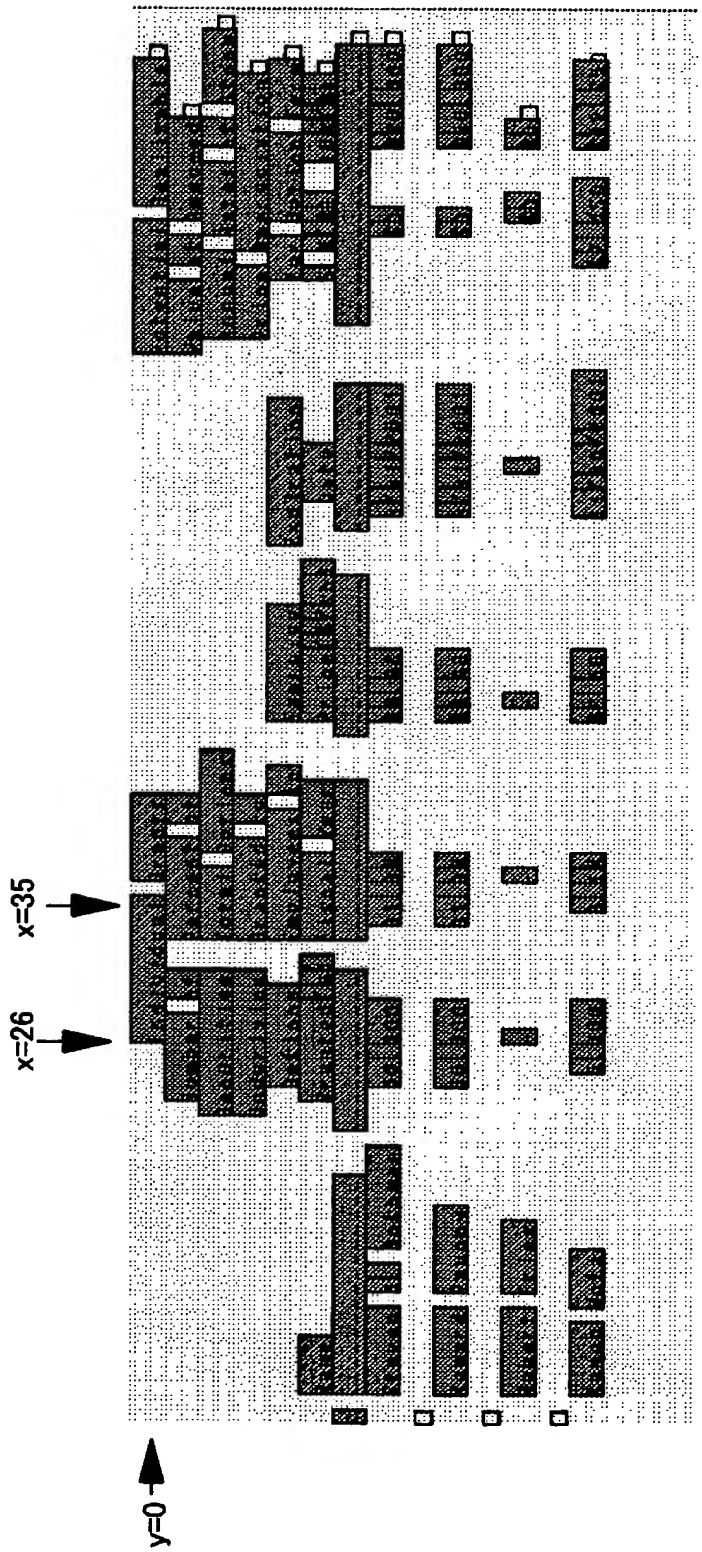
【図８】



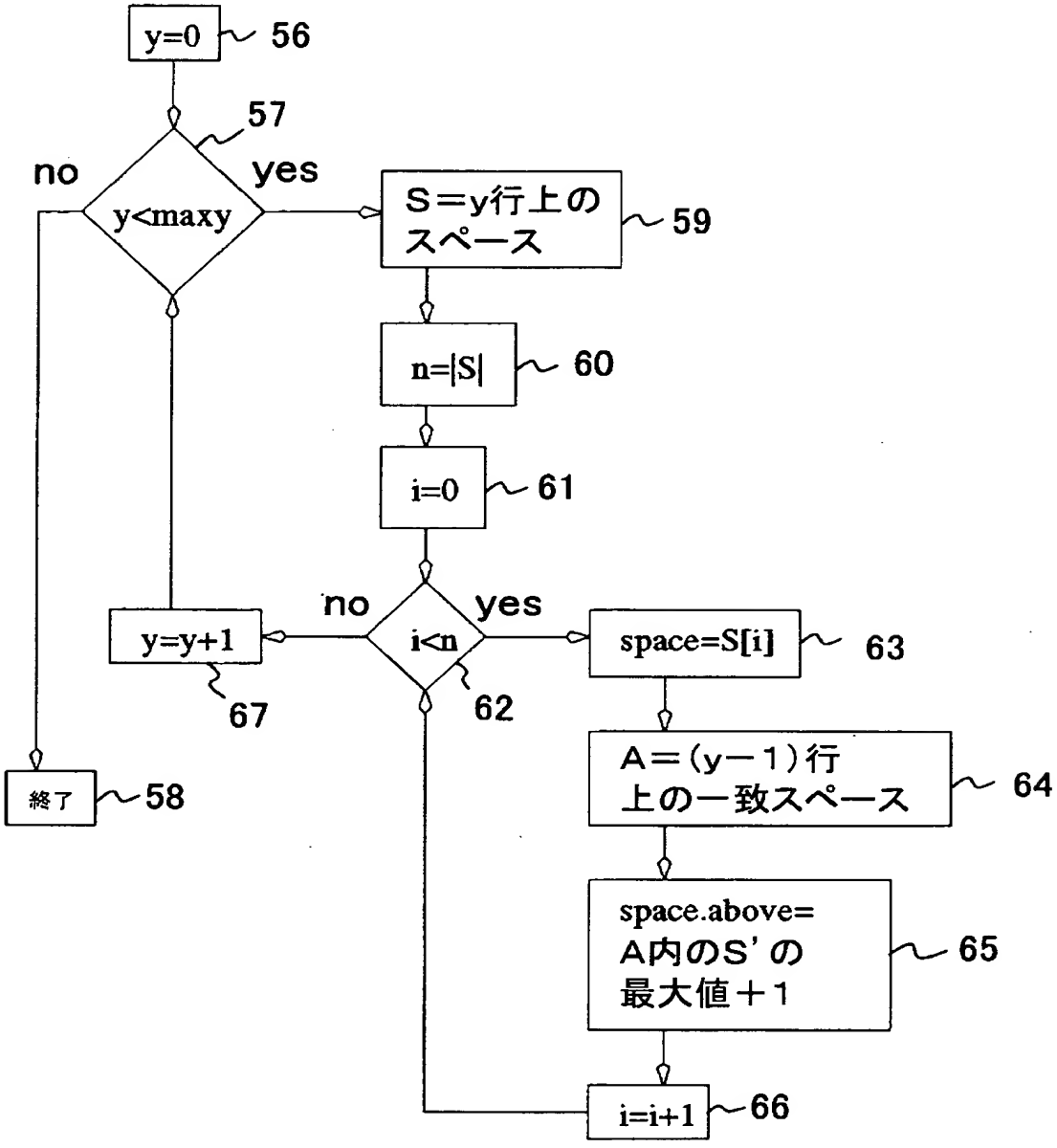
【図９】


開始
40
y=0
41
y<maxy
42
yes
no
終了
43
x=0
44
start=x
45
x<maxx
46
yes
no
y=y+1
47
T=char_type(x, y)
48
char_type(x, y) = T
49
yes
no
x<maxx
51
yes
no
x=x+1
50
token = (start, y, x-1, y)
52
T==space?
53
yes
no
スペースデータベースへのtokenの追加
54
トークンデータベースへのtokenの追加
55

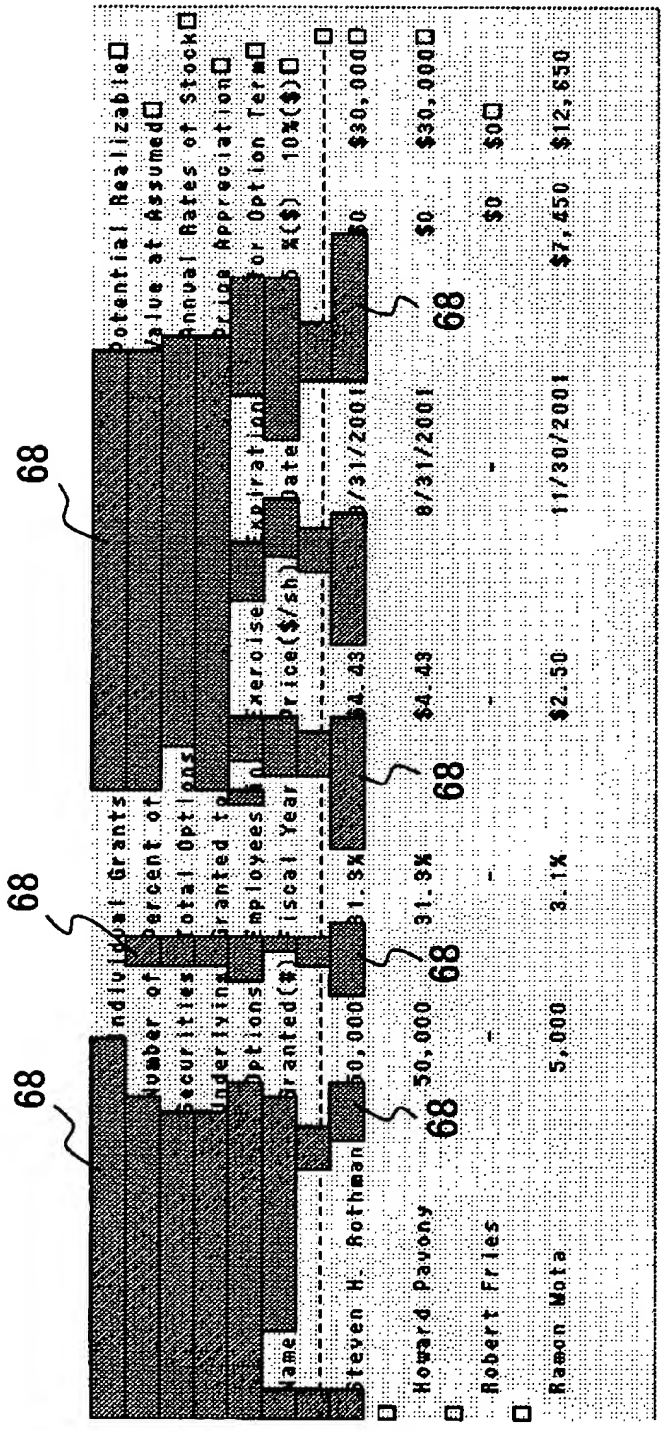
【図１０】



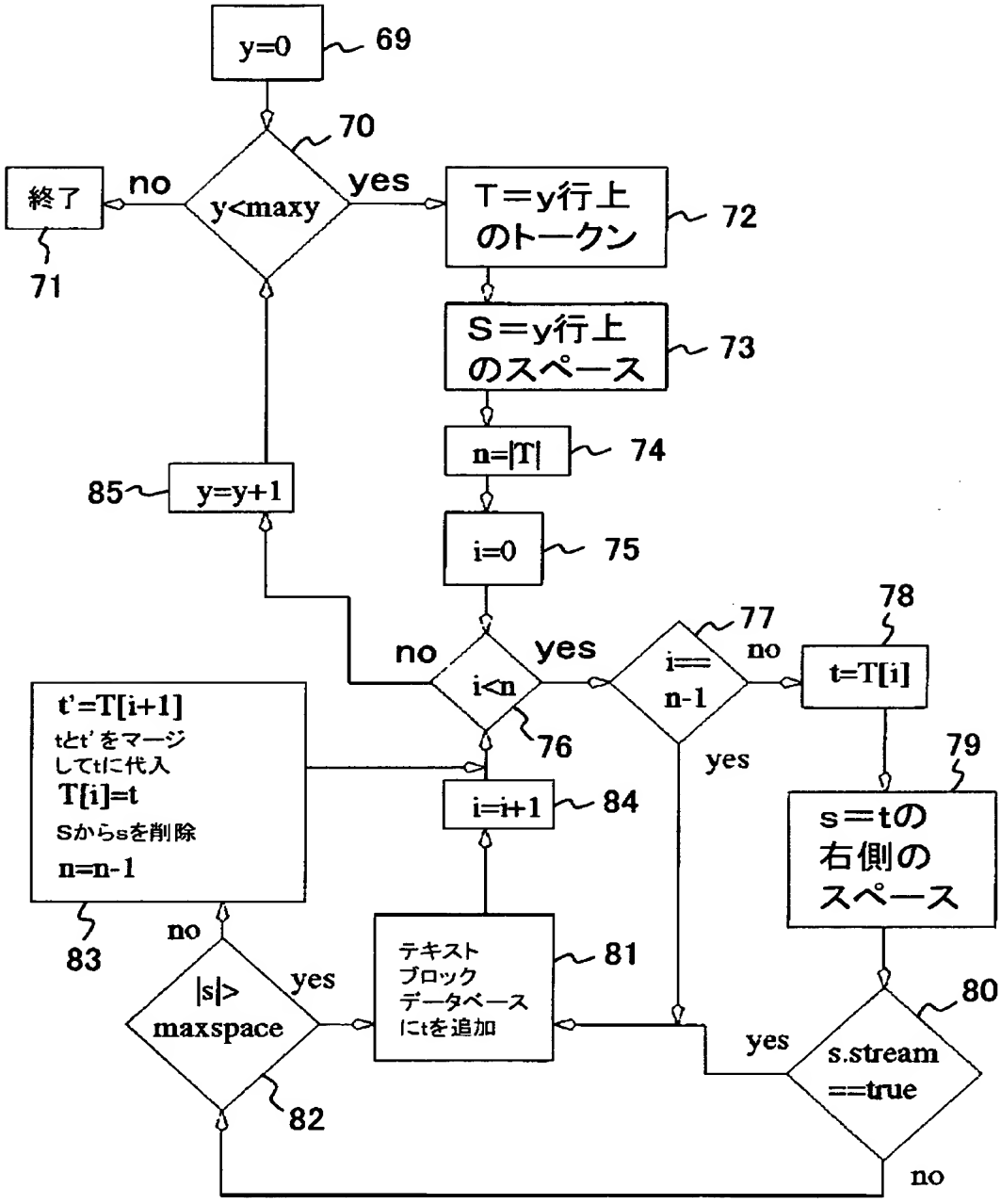
【図１１】


y=0
56
57
no
yes
y<maxy
S＝y行上の
スペース
59
n=|S|
60
i=0
61
no
yes
i<n
62
y=y+1
67
space=S[i]
63
終了
58
A＝(y－1)行
上の一致スペース
64
space.above=
A内のS'の
最大値＋1
65
i=i+1
66

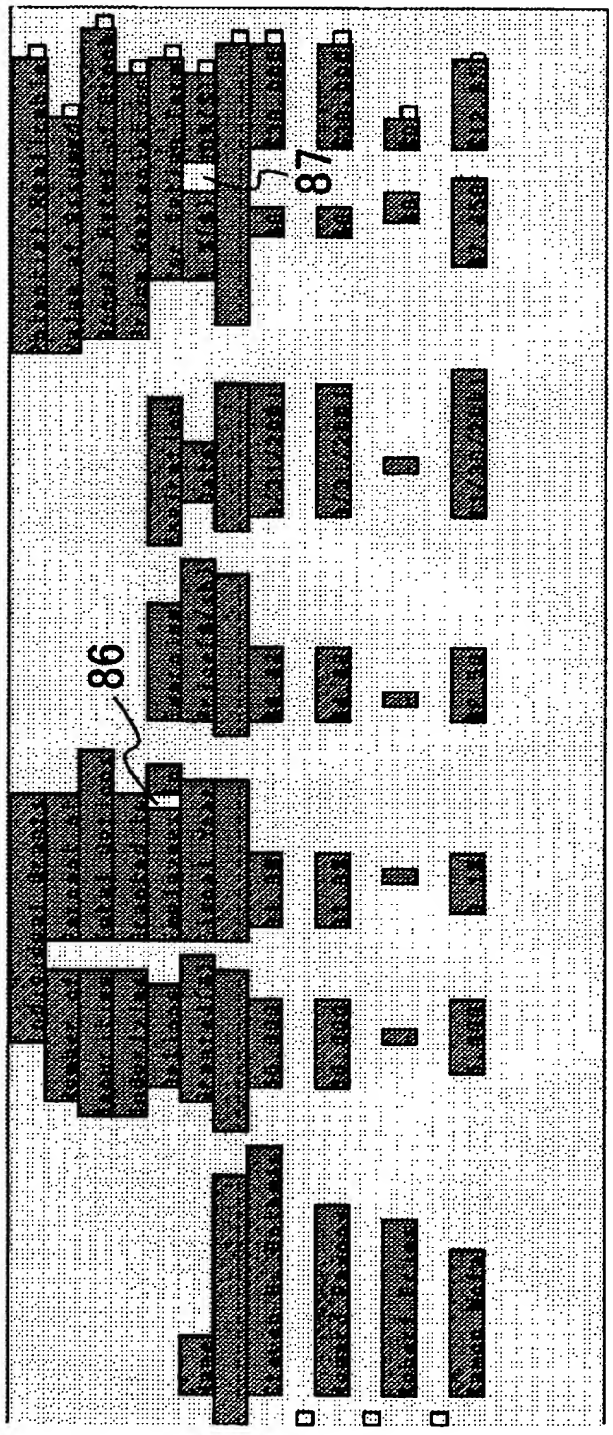
【図１２】


68
68
68
68
68
68
68
Name
Number of Securities Underlying Options Granted(#)
Individual Grants
Percent of Total Options Granted to Employees in Fiscal Year
Exercise Price($/sh)
Expiration Date
Potential Realizable Value at Assumed Annual Rates of Stock Price Appreciation for Option Term
5%($)
10%($)
Steven H. Rothman
50,000
31.3%
$4.43
8/31/2001
$0
$30,000
Howard Pavony
50,000
31.3%
$4.43
8/31/2001
$0
$30,000
Robert Fries
-
-
-
-
$0
$0
Ramon Mota
5,000
3.1%
$2.50
11/30/2001
$7,450
$12,650

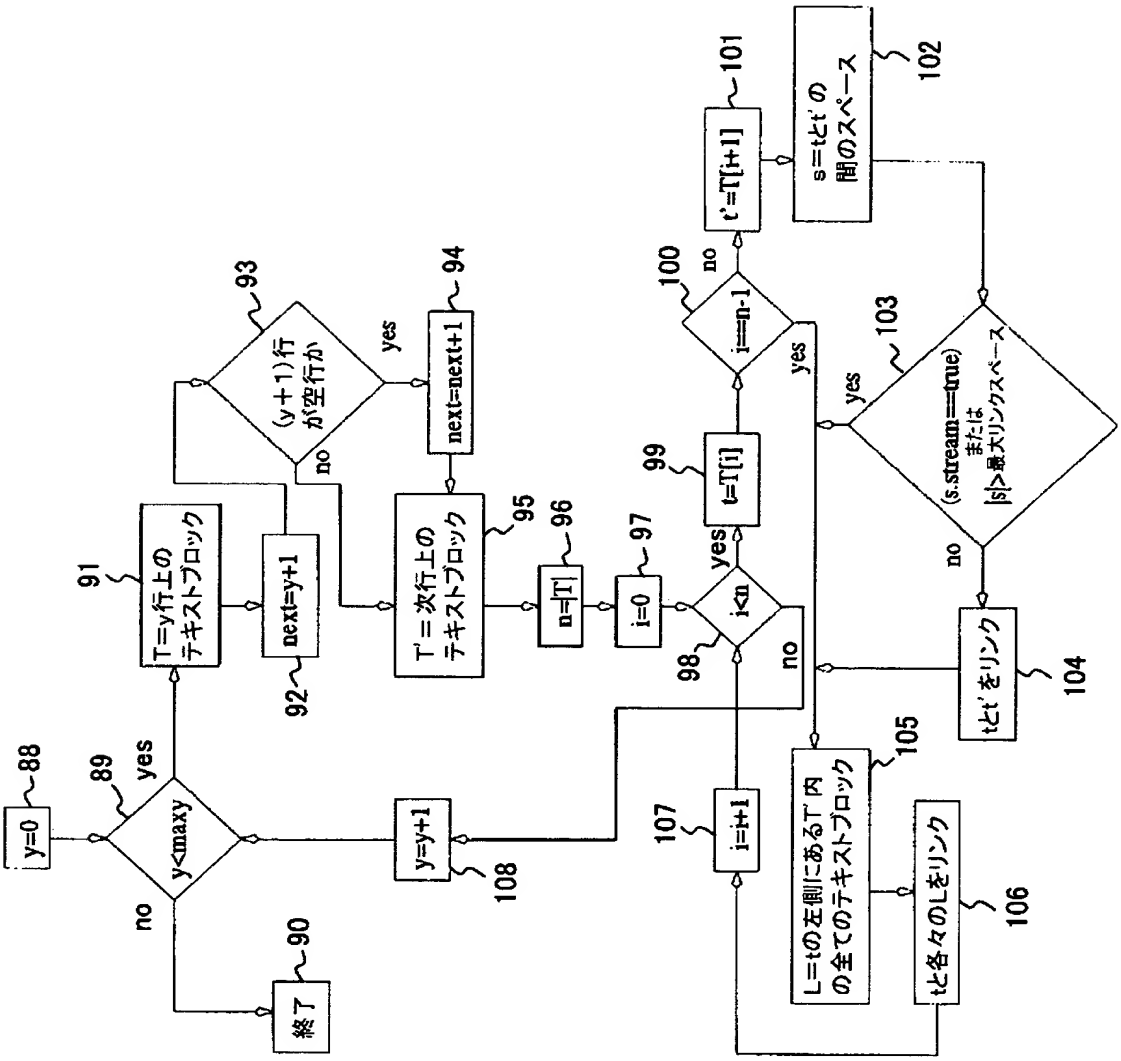
【図１３】


y=0
69
70
終了
no
y<maxy
yes
T＝y行上のトークン
72
71
S＝y行上のスペース
73
n=|T|
74
85
y=y+1
i=0
75
78
77
no
yes
i==n-1
no
t=T[i]
i<n
76
t'=T[i+1]
tとt'をマージしてtに代入
T[i]=t
Sからsを削除
n=n-1
yes
79
i=i+1
84
s＝tの右側のスペース
83
no
テキストブロックデータベースにtを追加
81
|s|>maxspace
yes
80
yes
s.stream==true
82
no

【図１４】



【図１５】


y=0
88
89
yes
y<maxy
no
終了
90
T=y行上のテキストブロック
91
next=y+1
92
(y+1)行が空行か
93
yes
no
next=next+1
94
T'=次行上のテキストブロック
95
n=|T|
96
i=0
97
i<n
98
yes
no
t=T[i]
99
i==n-1
100
no
yes
t'=T[i+1]
101
s=tとt'の間のスペース
102
(s.stream==true)
または
|s|>最大リンクスペース
103
yes
no
tとt'をリンク
104
L=tの左側にあるT'内の全てのテキストブロック
105
tと各々のLをリンク
106
i=i+1
107
y=y+1
108

【図１６】

```
1 (tokens, spaces)←tokenize(doc);
2 streams←stream(doc);
3 text_blocks←get_initial_blocks(tokens, spaces, streams);
4 links←get_initial_links(text_blocks);
5 document_graph←(text_blocks, links);
```


出証特２０００－３１０７２８７

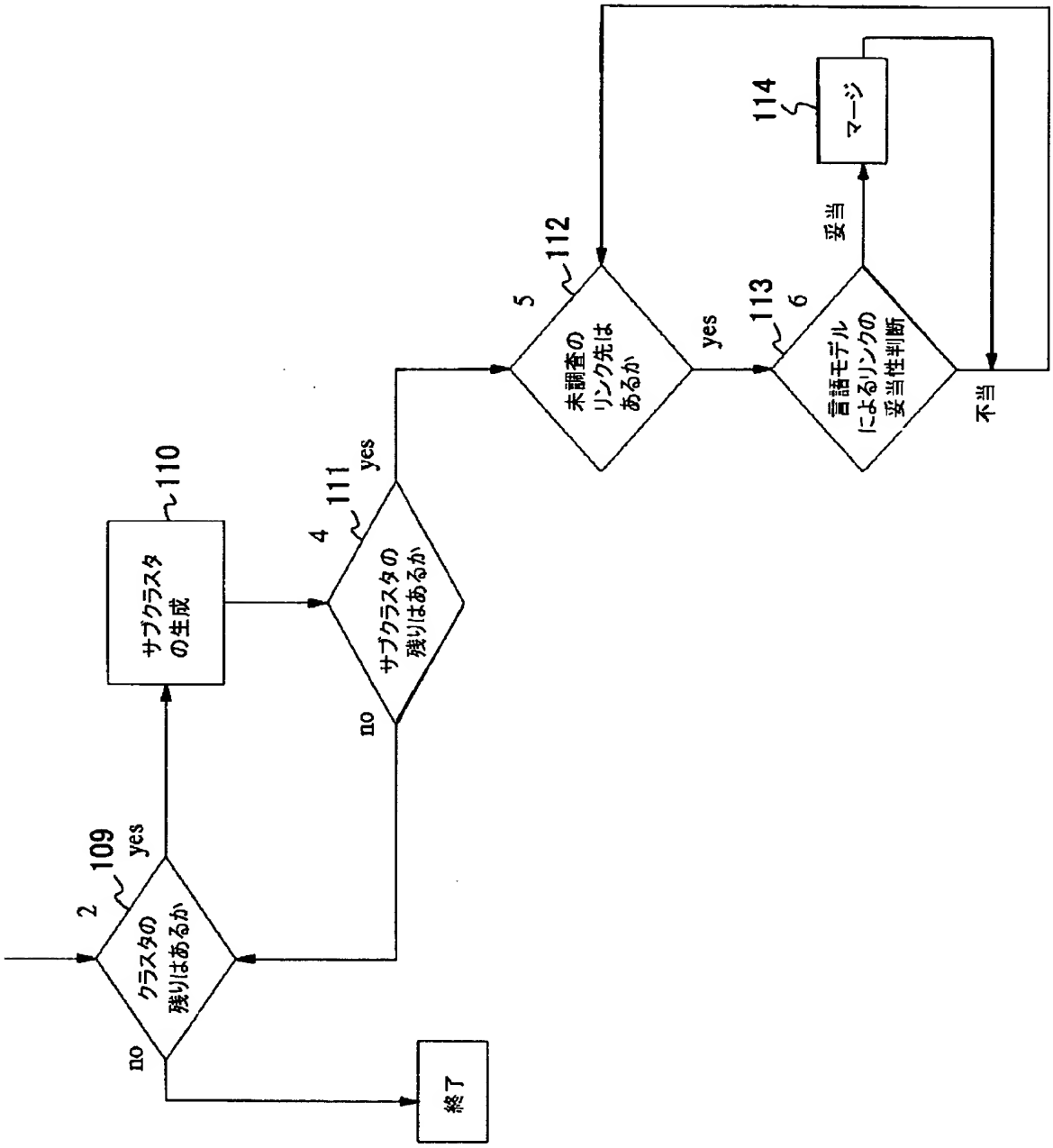
【図17】

```
1 cluster_set←cluster(doc);
2 for all c∈ cluster_set do {
3     sub_cluster_set←sub-cluster(c);
4     for all s∈ sub_cluster_set do {
5           for all links in s do {
6                 if valid(link) then merge(sink, source);
7           }
8     }
9 }
```


特2000-190335

出証特2000-3107287

【図１８】


クラスタの残りはあるか
yes
no
2
109
サブクラスタの生成
110
サブクラスタの残りはあるか
111
4
yes
no
未調査のリンク先はあるか
112
5
yes
言語モデルによるリンクの妥当性判断
113
6
妥当
不当
マージ
114
終了

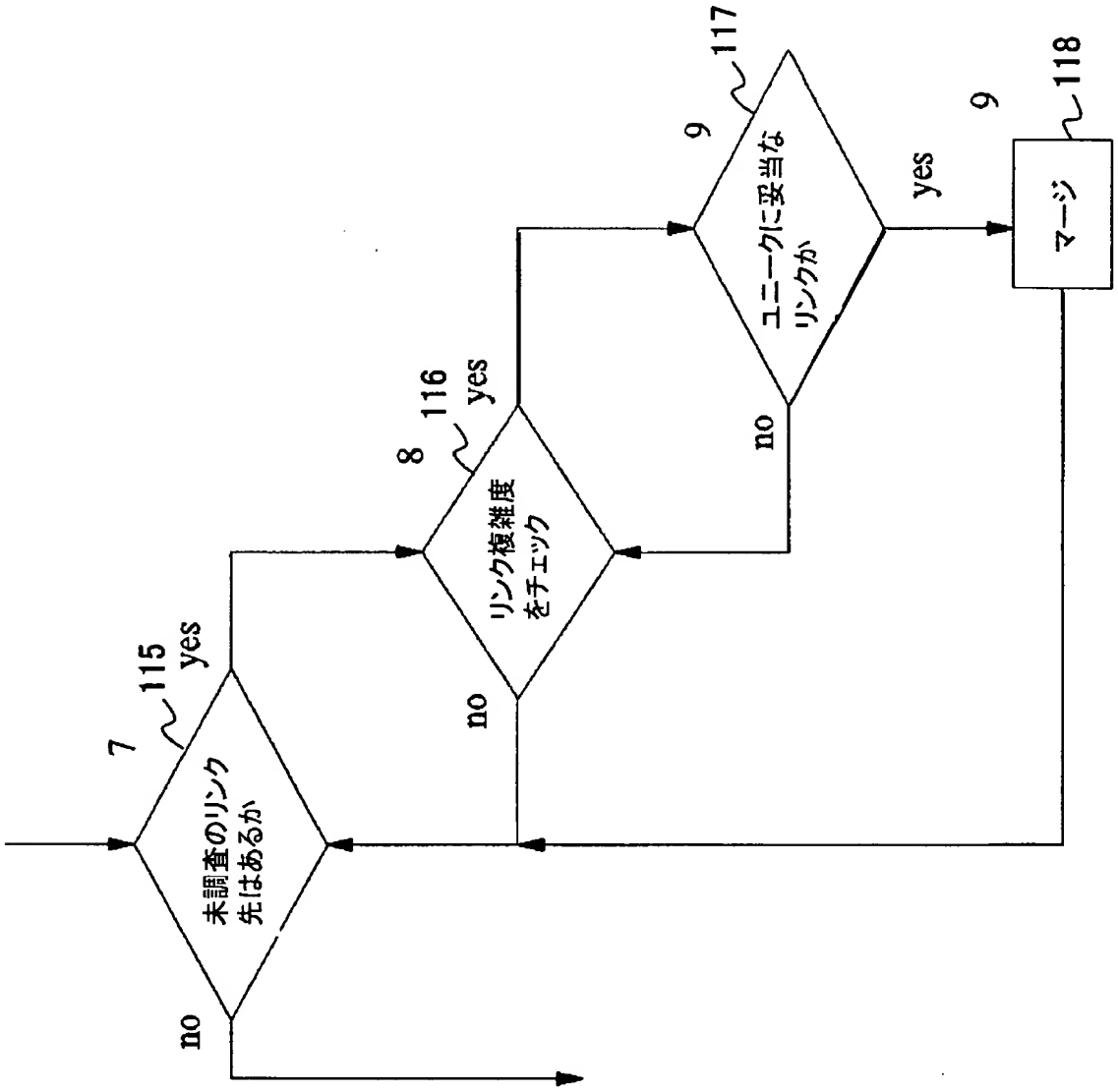
【図19】

```
1 perplexity←3;
2 max_perplexity←get_max_perplexity;
3 while(perplexity<max_perplexity) do {
4     repeat while merges continue to be carried out
5     cluster_set←cluster(doc);
6     for all c∈ cluster_set do {
7          for all links in c do {
8               if(perplexity(link)<perplexity) then do {
9                    if unique_valid_link(link) then merge(sink, source);
10              }
11         }
12    }
13    perplexity←perplexity + 1;
14    max_perplexity←get_max_perplexity;
15}
```


特2000−190335

出証特2000−3107287

【図２０】

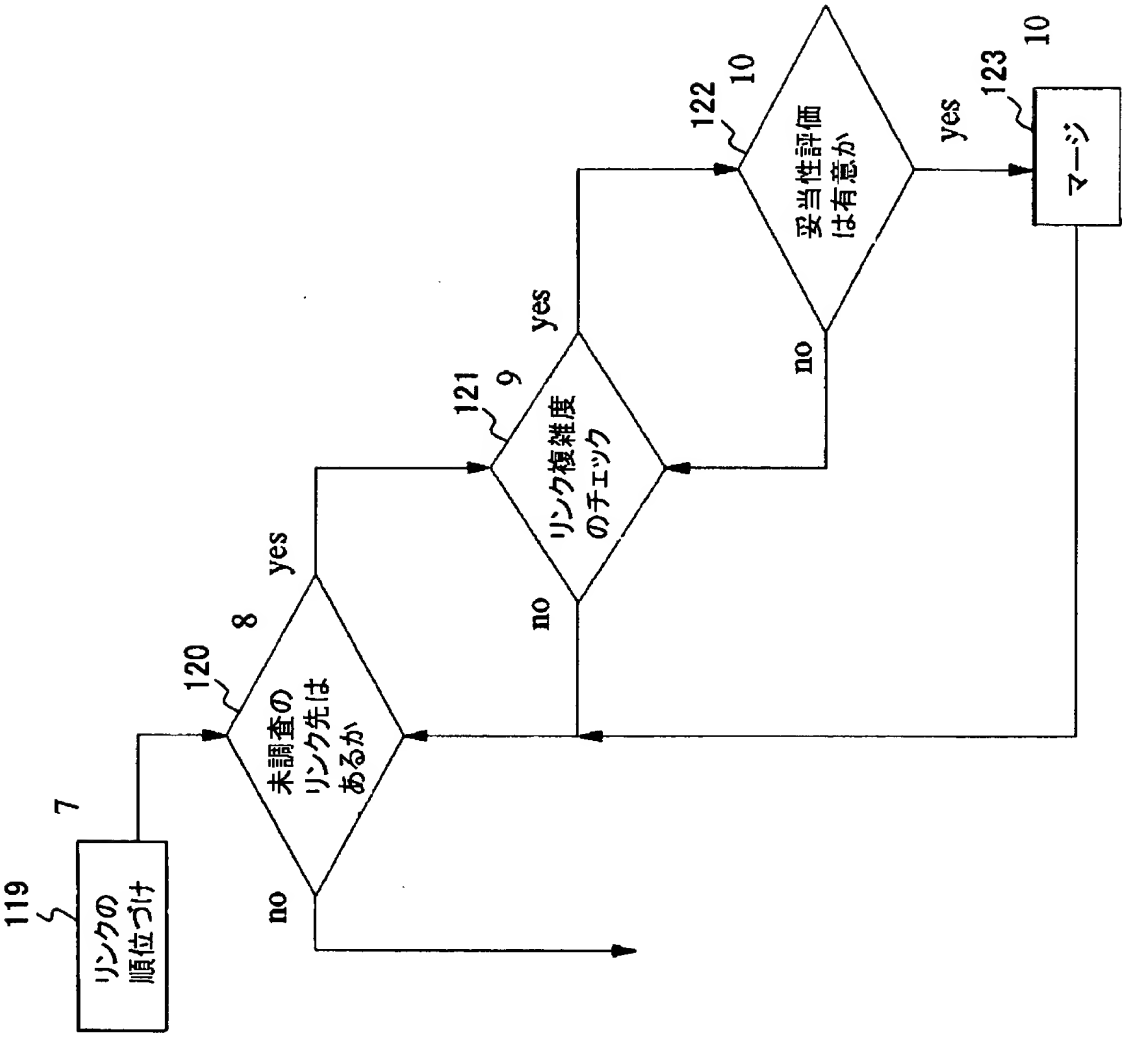

7
115
yes
未調査のリンク
先はあるか
no
8
116
yes
リンク複雑度
をチェック
no
9
117
ユニークに妥当な
リンクか
yes
9
118
マージ

【図21】

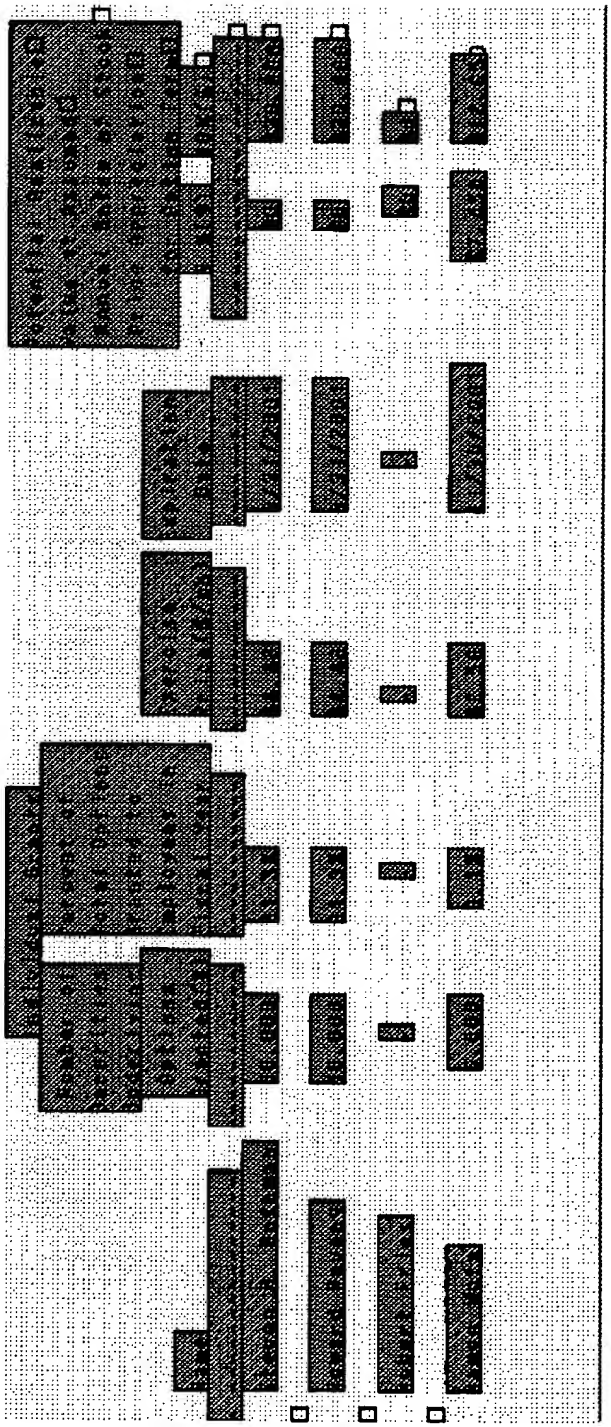
```
1 perplexity←3;
2 max_perplexity←get_max_perplexity;
3 while(perplexity<max_perplexity) do {
4     repeat while merges continue to be carried out
5     cluster_set←cluster(doc);
6     for all c∈ cluster_set do {
7          ordered_links←get_ordered_links(c);
8          for each link ∈ ordered_links do {
9               if perplexity(link)<perplexity then do {
10                   if distinguished_valid_link(link) then merge(sink, source);
11              }
12         }
13    }
14    merge_unary_sub-clusters;
15    perplexity←perplexity + 1;
16    max_perplexity←get_max_perplexity;
17}
```


出証特2000-3107287

【図２２】


119
リンクの順位づけ
7
120
未調査のリンク先はあるか
8
yes
no
121
リンク複雑度のチェック
9
yes
no
122
妥当性評価は有意か
10
yes
no
123
マージ
10

【図２３】



【図２４】

Double Column
False White Space Rivers
Apposed/ Marginal Material
Simple Apposed/Marginal Material
Unmarked Headings
Double Spacing
Elliptical Lists
Short Paragraphs
Multi-column cell
Multi-row Cell
Multi-row Stub Cells
Multiple Cells (below header)
Elliptical Cell Contents
Grid Quantisation
Orientation Variation
One to Many, Many to One

【書類名】　　要約書

【要約】

【課題】　表、箇条書き、多段組等任意にレイアウトされた文書から意味のあるテキストブロックを抽出する。

【解決手段】　空白等でレイアウトされた文書を入力し、文書の空間座標で関連付けたシンボルを取得する。シンボルから同一タイプのキャラクタの連続を抽出しトークンとスペースを生成する。列方向に連続したスペースからストリームを生成し、ストリームとトークンからテキストブロックを生成する。テキストブロック間のリンクを生成して、文書グラフとする。文書グラフ内のテキストブロック間の接続（リンク）の妥当性を言語モデルを用いて評価し、接続が妥当な場合はそのテキストブロックをマージする。

【選択図】　図８

出証特２０００－３１０７２８７

認定・付加情報

特許出願の番号	特願２０００－１９０３３５
受付番号	５００００７９２２６１
書類名	特許願
担当官	塩崎　博子　　１６０６
作成日	平成１２年　８月　４日

＜認定情報・付加情報＞

【特許出願人】

【識別番号】　390009531

【住所又は居所】　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　坂口　博

【復代理人】　申請人

【識別番号】　100112520

【住所又は居所】　神奈川県大和市中央林間３丁目４番４号　サクライビル４階　間山・林合同技術特許事務所

【氏名又は名称】　林　茂則

【選任した代理人】

【識別番号】　100091568

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　大和事業所内

【氏名又は名称】　市位　嘉宏

【選任した復代理人】

【識別番号】　100110607

【住所又は居所】　神奈川県大和市中央林間３丁目４番４号　サクライビル４階　間山・林合同技術特許事務所

【氏名又は名称】　間山　進也

【選任した復代理人】

【識別番号】　100098121

次頁有

認定・付加情報（続き）

【住所又は居所】　神奈川県大和市中央林間３丁目４番４号　サクライビル４階　間山・林合同技術特許事務所
【氏名又は名称】　間山　世津子

次頁無

出　願　人　履　歴　情　報

識別番号　　〔３９０００９５３１〕

１．変更年月日　　２０００年　５月１６日
［変更理由］　名称変更
住　所　　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）
氏　名　　インターナショナル・ビジネス・マシーンズ・コーポレーション